



TUGAS AKHIR - KI091391

Aplikasi Perangkat Bergerak Pencarian Wisata Berdasarkan Daerah Berbasis Android

Ahmad Sodik
NRP 5110 100 125

Dosen Pembimbing I
Dwi Sunaryono, S.Kom., M.Kom.

Dosen Pembimbing II
Ahatius Solichah Ahmadiyah, S.Kom., M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2015



FINAL PROJECT - KI091391

Android-based Mobile Application For Location-based Tours Searching

Ahmad Sodik
NRP 5110 100 125

Advisor I
Dwi Sunaryono, S.Kom., M.Kom.

Advisor II
Adhatus Solichah Ahmadiyah, S.Kom., M.Sc.

INFORMATICS DEPARTMENT
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2015

LEMBAR PENGESAHAN

**Aplikasi Bergerak Pencarian Wisata Berdasarkan Daerah
Berbasis Android**

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Rekayasa Perangkat Lunak
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

AHMAD SODIK
NRP : 5110 100 125

Disetujui oleh Dosen Pembimbing Tugas Akhir :

DWI SUNARYONO, S.Kom., M.Kom.

NIP: 197205281997021001

(pembimbing 1)

ADHATUS SOLICHAH AHMADIYAH,

S.Kom., M.Sc.

NIP: 5100201405002

(pembimbing 2)

**SURABAYA
JUNI 2015**

Aplikasi Bergerak Pencarian Wisata Berdasarkan Daerah Berbasis Android

Nama Mahasiswa : Ahmad Sodik
NRP : 5110100125
Departement : Teknik Informatika FTIf-ITS
Dosen Pembimbing 1 : Dwi Sunaryono, S.Kom., M.Kom.
Dosen Pembimbing 2 : Adhatus Solichah A, S.Kom., M.Sc.

ABSTRAK

Perkembangan sistem operasi Android yang sangat pesat pada saat ini, banyak dimanfaatkan untuk mengembangkan aplikasi yang bermanfaat bagi manusia seperti untuk kesehatan, wisata, game dan lain – lain.

Aplikasi informasi wisata merupakan salah satu aplikasi yang mulai banyak dikembangkan pada sistem operasi Android. Aplikasi berjenis ini dapat memberikan informasi kepada masyarakat tentang banyaknya objek dan daya tarik wisata. Di provinsi Jawa Timur ada beraneka ragam objek dan daya tarik wisata, dimana setiap bagian daerah memiliki ciri khas wisata masing-masing. Umumnya masyarakat hanya mengetahui wisata-wisata yang sudah populer, padahal ada banyak objek dan daya tarik wisata yang belum populer dan tidak kalah menarik untuk dikunjungi. Hal ini karena kurangnya penyedia informasi tentang wisata daerah di provinsi Jawa Timur. Dari aplikasi wisata yang sudah ada saat ini, diperlukan rekomendasi wisata untuk memberikan pilihan wisata dan menginformasikan adanya wisata lain yang sejenis.

Untuk mengatasi permasalahan tersebut, pada Tugas Akhir ini dikembangkan aplikasi perangkat bergerak pencarian tempat wisata berdasarkan daerah berbasis android (WisataINA) adalah aplikasi mobile pada sistem operasi Android yang menyediakan informasi tentang tempat-tempat wisata daerah di provinsi Jawa Timur. Aplikasi WisataINA menyediakan informasi tentang setiap wisata berupa judul, alamat sumber, ulasan dan foto wisata, aplikasi ini juga menyediakan fitur penunjuk jalan ke tempat wisata menggunakan Google Maps (peta) yang fungsinya untuk mempermudah wisatawan mencari jalan ke tempat wisata. Selain fitur tersebut, aplikasi WisataINA juga memiliki fitur

rekomendasi wisata yang hampir sama dengan wisata yang dicari oleh pengguna dan fitur pencarian wisata terdekat dengan lokasi pengguna. Dalam aplikasi WisataINA, wisata dibagi menjadi kategori alam, sejarah, belanja, kuliner, budaya dan religi. Sedangkan untuk kriteria biaya, dibagi menjadi kriteria murah (Rp.0-Rp.50000), sedang (Rp.50001-Rp.150000), dan mahal (lebih dari Rp.150000).

Untuk pengujian pada aplikasi WisataINA digunakan metode pengujian kotak hitam (black box) dan pengujian kegunaan. Pengujian kotak hitam adalah pengujian setiap fitur untuk melihat hasil yang didapat. Pengujian kotak hitam digunakan untuk menguji semua fungsionalitas aplikasi apakah berjalan sesuai dengan prosesnya. Pada pengujian kotak hitam semua fungsionalitas berjalan dengan baik. Pengujian kegunaan digunakan untuk mengukur tingkat kegunaan aplikasi pada pengguna, dari hasil pengujian aplikasi ke beberapa orang didapatkan nilai di atas 75% untuk setiap aspek. Maka dapat disimpulkan Aplikasi WisataINA mendapatkan hasil lebih dari cukup.

Kata Kunci Aplikasi Perangkat Bergerak Pencarian Wisata, Wisata Daerah Jawa Timur.

Android-based Mobile Application For Location-based Tours Searching

Student Name : Ahmad Sodik
NRP : 5110100125
Jurusan : Teknik Informatika FTIf-ITS
Advisor 1 : Dwi Sunaryono, S.Kom., M.Kom.
Advisor 2 : Adhatus Solichah Ahmadiyah, S.Kom.,
M.Sc.

ABSTRACT

Nowadays, the rapid development of android operating system used to develop Android-based mobile applications which is beneficial for humans such as for healths, travels, games etc.

The information tour application is one of application that begin to develop on Android operating system. This type of application can give information the users about many object and attractions tours. The Province of East Java offers many variety of object and Tourism place in each districts, each districts has special characteristics kind of tourism place. Usually many people, only know the tourism place that already popular, whereas unknown many object and attractions of tourism are quite interesting to visit. The condition caused by less of information provider about tourism place in East Java. Out of many available tour applications which already exist, needs the tourism recommendations to give choice and to inform there are another tourism place which almost same genre.

To solve those problems mentioned above, on this final assignment the developing Android-based Mobile Applications for Location-based Tours Searching (WisataINA) is mobile application on android system operation which provide the information about Tourism Places in East Java. The WisataINA application provide the information about every Tourism place with title, address resources, reviews and Tourism image, this application also provides pointing way futures to easily the Tourist to find the way to Tourism location. Beside those futures mention, the WisataINA application also has Tourism recommendation future which almost same with Tourism place that looked

by user and nearest tour search with user location. In WisataINA application, tourism divided into categories of nature, history, shopping, culinary, culture and religion. Meanwhile, the criteria of cost divided from cheaper cost (Rp.0-5000), medium (Rp.50001-150000), and expensive (up Rp.150000).

To test the WisataINA application used the blackbox testing and usability testing method. The blackbox testing is testing every features to see the final result. The Blackbox testing used to test all application functionality whether or not the output satisfied our desire. In this testing we got the functionality already satisfied our desire. Usability testing used to test usability level againsts user. In this testing we got above 75% value for every aspect. That indicates WisataINA application already give above average result.

Keyword: Mobile Applications Tours Search, Location-based East Jawa Tours.

KATA PENGANTAR

Puji syukur kepada Allah Yang Maha Esa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan tugas akhir yang berjudul:

Aplikasi *Mobile* Pencarian Tempat Wisata Berdasarkan Daerah Berbasis Android

Melalui lembar ini, penulis hanya ingin menyampaikan ucapan terima kasih dan penghormatan yang sebesar-besarnya kepada:

1. Allah SWT atas segala nikmat dan rahmat yang telah diberikan selama ini.
2. Ayah, Ibu dan keluarga penulis yang tiada henti-hentinya mencurahkan kasih sayang, perhatian, dan doa kepada penulis selama ini.
3. Bpk Dwi Sunaryono, S.Kom., M.Kom. dan Ibu Adhatus Solichah Ahmadiyah, S.Kom., M.Sc. selaku dosen pembimbing yang telah memberikan bimbingan, motivasi, dan meluangkan waktu untuk membantu pengerjaan Tugas Akhir ini.
4. Bapak dan Ibu dosen Teknik Informatika ITS yang telah membina dan memberikan ilmu kepada penulis selama menempuh studi di Teknik Informatika ITS.
5. Orang tua yang selama ini sudah mendukung dan mengingatkan untuk segera menyelesaikan tugas akhir ini.
6. Hizkia Adi Surya, Farid Naufal, Dimas Gesang, Yohanes Surya, Arie Priyambada, Ayu Wandira, Galuh Eva Nanda, Agus Triwibowo dan Sahabat dekat yang selalu memberikan dukungan dan mengingatkan.

7. Teman-teman yang ada di laboratorium GCL yang selalu mengingatkan tentang Tugas Akhir ketika lupa.

Penulis sudah berusaha sebaik-baiknya dalam menyelesaikan Tugas Akhir ini, namun apabila terdapat kekurangan dalam penyusunannya penulis memohon maaf sebesar-besarnya.

Surabaya, Juni 2015

Ahmad Sodik

DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xxi
DAFTAR TABEL	xxv
DAFTAR KODE SUMBER	xxvii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan.....	2
1.3. Rumusan Permasalahan	2
1.4. Batasan Permasalahan	3
1.5. Metodologi	3
1.6. Sistematika Penulisan	5
BAB II DASAR TEORI.....	7
2.1. Aplikasi wisata yang sudah ada.....	7
2.2. Pariwisata Jawa Timur.....	7
2.3. <i>JavaScript Object Notation</i>	9
2.2.1. Protokol Web (SOAP dan REST).....	10
2.2.2. MySQL.....	10
2.1.3. <i>Website Scraping</i>	11
2.4. <i>Analytic Hierarchy Process</i> (AHP)	12
2.5. Metode Haversine.....	13

2.6.	Sistem Operasi Android.....	13
2.7.	Android Global Positioning System (GPS).....	14
2.8.	Google Maps	15
2.9.	Google Places.....	16
BAB III ANALISIS DAN PERANCANGAN SISTEM.....		19
3.1.	Analisis	19
3.1.1.	Analisis Permasalahan	19
3.1.1.1.	Analisis permasalahan mendapatkan data dengan <i>scraping website</i>	23
3.1.1.2.	Analisis permasalahan penerapan algoritma Analitic HierarchiAHP	24
3.1.1.3.	Analisis permasalahan penerapan metode Haversine	29
3.1.1.4.	Analisis permasalahan penunjuk jalan ke lokasi wisata dengan Google Maps	29
3.1.2.	Deskripsi Umum Sistem	30
3.1.3.	Aktor	31
3.1.4.	Kebutuhan Fungsional	31
3.1.5.	Kebutuhan Non Fungsional.....	32
3.1.6.	Kasus Penggunaan	32
3.1.6.1.	Mengelola Kota.....	33
3.1.6.2.	Mengelola Kategori.....	36
3.1.6.3.	Mengelola Wisata	38
3.1.6.4.	Mencari Wisata	40
3.1.6.5.	Melihat Detail Wisata.....	41
3.1.6.6.	Menggunakan Penunjuk Jalan	43
3.1.7.	Diagram Arsitektur Aplikasi	44

3.2.	Perancangan Basis Data.....	47
BAB IV IMPLEMENTASI.....		51
4.1.	Lingkungan Pembangunan Perangkat Lunak.....	51
4.2.	Implementasi Antarmuka Pengguna Admin	51
4.2.1.	Antarmuka Administrator <i>Login</i>	52
4.2.2.	Antarmuka Administrator Melihat Wisata	52
4.2.3.	Antarmuka Administrator Tambah Wisata Manual 53	
4.2.4.	Antarmuka Administrator Tambah Wisata dengan <i>Scraping Website</i> lain.	55
4.2.5.	Antarmuka Administrator Mengelola Kota.....	56
4.2.6.	Antarmuka Administrator Mengelola Kategori...57	
4.2.7.	Antarmuka Administrator Mengelola Wisata.....	58
4.3.	Implementasi Antarmuka Pengguna Aplikasi.....	59
4.3.1.	Antarmuka Menu Aplikasi	59
4.3.2.	Antarmuka Menu Pencarian	60
4.3.3.	Antarmuka Pencarian Berdasarkan Kriteria.....	61
4.3.4.	Antarmuka Hasil Pencarian Berdasarkan Kriteria 62	
4.3.5.	Antarmuka Pencarian Cepat	63
4.3.6.	Antarmuka Detail Wisata	64
4.3.7.	Antarmuka Penunjuk Jalan.....	65
4.3.8.	Antarmuka Pencarian Terdekat	66
4.4.	Implementasi Proses	67
4.4.1.	Implementasi Basis Data.....	68
4.4.2.	Implementasi <i>Query</i>	70

4.4.2.1.	Implementasi <i>Query</i> Pencarian Berdasarkan Kriteria	70
4.4.2.2.	Implementasi <i>Query</i> Pencarian Cepat dengan Rekomendasi.....	70
4.4.2.3	Implementasi <i>Query</i> Detail Wisata	71
4.4.2.4	Implementasi <i>Query</i> Pencarian Terdekat	71
4.4.3	Impelementasi <i>Webservice</i>	72
4.4.3.1	Implementasi <i>Webservice</i> Pencarian Berdasarkan Kriteria.....	72
4.4.3.2	Implementasi <i>Webservice</i> Pencarian Cepat dengan Rekomendasi.....	74
4.4.3.3	Implementasi <i>Webservice</i> Detail Wisata	76
4.4.3.4	Implementasi <i>Webservice</i> Pencarian Terdekat	77
4.4.4	Implementasi Perangkat Bergerak (<i>Mobile</i>).....	78
4.4.5	Implementasi Aplikasi <i>Website</i>	85
BAB V PENGUJIAN DAN EVALUASI		103
5.1	Lingkungan Pengujian	103
5.2.	Dasar Pengujian.....	103
5.3.	Pengujian Fungsional.....	104
5.3.1.	Pengujian Pengguna Aplikasi.....	104
5.3.1.1.	Pengujian Pencarian Berdasarkan Kriteria	104
5.3.1.2.	Pengujian Pencarian Cepat	106
5.3.1.3.	Pengujian Pencarian Wisata Terdekat	107
5.3.1.4.	Pengujian Menampilkan Detail Wisata	108
5.3.1.5.	Menampilkan Penunjuk Jalan	109
5.3.1.	Pengujian Pengguna Admin	111
5.3.2.1.	Melihat Wisata	111

5.3.2.2.	Tambah Wisata Manual.....	112
5.3.2.3.	Tambah Wisata Dengan <i>Scraping Website</i> Lain	114
5.3.2.4.	Mengelola Kota.....	116
5.3.2.5.	Mengelola Kategori.....	117
5.3.2.6.	Mengelola Wisata	118
5.4.	Pengujian Kegunaan.....	119
5.4.1.	Kriteria Responden	119
5.4.2.	Lingkungan Pengujian Kegunaan.....	119
5.4.3.	Skenario Pengujian Kegunaan.....	119
5.4.4.	Daftar Responden	120
5.4.5.	Hasil Pengujian Kegunaan	121
5.4.5.1	Rekapitulasi Pengujian Aplikasi.....	121
5.4.5.2	Evaluasi Pengujian Fungsionalitas	122
5.4.5.3	Evaluasi Pengujian Kegunaan	123
5.4.6.	Pengujian Perbandingan Kebutuhan Fungsional dengan Aplikasi WisataLokal dan Aplikasi Wisataku	124
BAB VI KESIMPULAN DAN SARAN.....		131
6.1.	Kesimpulan	131
6.2.	Saran	132
LAMPIRAN A – FORMULIR PENILAIAN PENGUJIAN KEGUNAAN		135
LAMPIRAN B – IMPLEMENTASI SURVEI KRITERIA BIAYA.....		137
LAMPIRAN C – IMPLEMENTASI SURVEI PRIORITAS KATEGORI/BIAYA MASUK.....		139
BIODATA PENULIS		141

DAFTAR TABEL

Tabel 3.1. Kriteria Kategori	25
Tabel 3.2. Kriteria Biaya Masuk.....	25
Tabel 3.3. Skala Saaty.....	25
Tabel 3.4. Matrik Perbandingan	26
Tabel 3.5. Matrik Perbandingan Penyederhanaan	27
Tabel 3.6. Matrik Perbandingan Normalisai.	27
Tabel 3.7. Tabel Inkonsistensi Rasio	28
Tabel 3.8. Daftar Kebutuhan Fungsional Perangkat Lunak	31
Tabel 3.9. Daftar Kebutuhan Non Fungsional Perangkat Lunak.....	32
Tabel 3.10. Daftar Kode Diagram Kasus Penggunaan.....	33
Tabel 3.11. Spesifikasi Kasus Mengelola Kota	34
Tabel 3.12. Spesifikasi Kasus Mengelola Kategori	36
Tabel 3.12. Spesifikasi Kasus Mengelola Tempat Wisata	38
Tabel 3.13. Spesifikasi Kasus Mencari Wisata.	40
Tabel 3.14. Spesifikasi Kasus Melihat Dateil Wisata.	41
Tabel 3.15. Spesifikasi Kasus Penggunaan Peta Wisata.....	43
Tabel 3.16. Spesifikasi Basis Data Pencarian Wisata	48
Tabel 5.1. Lingkungan Pengujian Perangkat Lunak	103
Tabel 5.2. Pengujian Fungsional	104
Tabel 5.3. Pengujian pencarian berdasarkan kriteria	105
Tabel 5.4. Pengujian Pencarian Cepat	106
Tabel 5.5. Pengujian Melakukan Pencarian Wisata Terdekat ...	107
Tabel 5.6. Pengujian Menampilkan Detail Wisata	109
Tabel 5.7. Pengujian Menampilkan Penunjuk Jalan	110

Tabel 5.8. Pengujian Melihat Wisata Administrator.....	111
Tabel 5.9. Pengujian Tambah Manual	112
Tabel 5.10. Pengujian Menambah Wisata dengan Scrapping ...	114
Tabel 5.11. Pengujian Mengelola Kota.....	116
Tabel 5.12. Pengujian Mengelola Kategori.....	117
Tabel 5.13. Pengujian Mengelola Wisata.....	118
Tabel 5.14. Daftar Responden Pengujian Kegunaan	120
Tabel 5.15. Daftar Perubahan Penilaian Pada Pengujian Kegunaan	121
Tabel 5.16. Penilaian Kebutuhan Fungsional.....	122
Tabel 5.17. Rekapitulasi Akhir Pengujian Kegunaan.....	123
Tabel 5.18. Skenario Perbandingan Pencarian	124
Tabel 5.19. Hasil Perbandingan Aplikasi.....	125

DAFTAR GAMBAR

Gambar 3.1. Alur proses pencarian aplikasi perangkat bergerak	22
Gambar 3.2. Hirarki permasalahan dalam algoritma AHP.....	24
Gambar 3.4. Diagram Kasus Penggunaan.....	33
Gambar 3.5. Diagram Aktivitas Mengelola Kota	35
Gambar 3.6. Diagram Aktivitas Mengelola Kategori	37
Gambar 3.7. Diagram Aktivitas Mengelola Wisata.....	39
Gambar 3.8. Diagram Aktivitas Mencari Wisata	41
Gambar 3.9. Diagram Aktivitas Melihat Detail Wisata	42
Gambar 3.10. Diagram Aktivitas Penggunaan Penunjuk Jalan. .	44
Gambar 4.11. Diagram Arsitektur Sistem WisataINA.....	45
Gambar 3.12. Daigram Aristektur Perangkat Lunak	46
Gambar 3.13. Conseptual Data Model Pencarian Wisata	47
Gambar 3.14. Physical Data Model Basis Data Obat	48
Gambar 4.1. Antarmuka Administrator <i>Login</i>	52
Gambar 4.2. Antarmuka Administrator Pencarian	53
Gambar 4.3. Antarmuka Tambah Wisata Manual	54
Gambar 4.4. Antarmuka Hasil Tambah Wisata Manual	54
Gambar 4.5. Antarmuka Tambah Wisata Dengan <i>Website Scraping</i>	55
Gambar 4.6. Antarmuka Hasil Tambah Wisata <i>Website Scraping</i>	56
Gambar 4.7. Antarmuka Admisnistrator Mengelola Kota	57
Gambar 4.8 Antarmuka Adminstrator Mengelola Kategori.....	58
Gambar 4.9 Antarmuka Administrator Mengelola Wisata.....	59
Gambar 4.10. Antarmuka Menu Aplikasi	60

Gambar 4.11. Antarmuka Menu Pencarian.....	61
Gambar 4.12. Antarmuka Pencarian Berdasarkan Kriteria.....	62
Gambar 4.13. Antarmuka Hasil Pencarian Berdasarkan Kriteria	63
Gambar 4.14. Antarmuka Pencarian Cepat.....	64
Gambar 4.15. Antarmuka Detail Wisata.....	65
Gambar 4.16. Antarmuka Penunjuk Jalan.....	66
Gambar 4.17 Antarmuka Pencarian Terdekat	67
Gambar 5.1 Halaman Pencarian Berdasarkan Kriteria	105
Gambar 5.2. Halaman Pencarian Cepat	107
Gambar 5.3. Halaman Pencarian Wisata Terdekat	108
Gambar 5.4. Halaman Detail Wisata	109
Gambar 5.5. Halaman antarmuka penunjuk jalan.....	110
Gambar 5.6. Halaman Melihat Wisata Administrator	112
Gambar 5.7. Halaman Tambah Wisata Manual.....	113
Gambar 5.8. Halaman Detail Wisata Tambah Manual	113
Gambar 5.9. Halaman Tambah Wisata Dengan <i>Scrapping Website</i> Lain	115
Gambar 5.11. Halaman Mengelola kota.....	116
Gambar 5.12. Halaman Mengelola Kategori.....	117
Gambar 5.13. Halaman Mengelola Wisata	118
Gambar 5.14. WisataINA Fitur Pencarian Sesuai Kriteria.....	125
Gambar 5.15. WisataINA Fitur Pencarian Cepat Dengan Rekomendasi	126
Gambar 5.16. WisataINA Fitur Pencarian Wisata Terdekat	126
Gambar 5.17. Wisataaku Fitur Pencarian	127
Gambar 5.18. WisataLokal Fitur Pencarian	127

Gambar 5.19. WisataINA Detail Wisata.....	128
Gambar 5.20. Wisataku Detail Wisata.....	128
Gambar 5.21. WisataLokal Detail Wisata.....	129
Gambar 5.22. WisataINA Fitur Penunjuk Jalan	129
Gambar 5.23. Wisataku Fitur Penunjuk Jalan	130
Gambar 6.1. Implementasi Survei Kriteria Biaya.....	137
Gambar 6.2. Survei Prioritas Kategori/Biaya Masuk.....	139

DAFTAR KODE SUMBER

Kode Sumber 4.1. Implementasi Table Admin.	68
Kode Sumber 4.2. Implementasi Table Catatan.	68
Kode Sumber 4.3. Implementasi Tabel Provinsi	68
Kode Sumber 4.4. Implementasi Tabel Kota	69
Kode Sumber 4.5. Implementasi Tabel Kategori	69
Kode Sumber 4.6. Implementasi Tabel Wisata	69
Kode Sumber 4.7. Implementasi <i>Query</i> Pencarian Wisata Berdasarkan Daerah.....	70
Kode Sumber 4.8. Implementasi <i>Query</i> Pencarian Cepat dengan Rekomendasi	71
Kode Sumber 4.9. Implementasi <i>Query</i> Detail Wisata	71
Kode Sumber 4.10. Implementasi <i>Query</i> Pencarian Terdekat.....	72
Kode Sumber 4.11. Implementasi <i>Webservice</i> Pencarian Berdasarkan Kriteria	74
Kode Sumber 4.12. Implementasi <i>Webservice</i> Pencarian Cepat Dengan Rekomendasi	76
Kode Sumber 4.13. Implementasi <i>Webservice</i> Detail Wisata	77
Kode Sumber 4.14. Implementasi <i>Webservice</i> Pencarian Terdekat	78
Kode Sumber 4.15. Implementasi Java Pencarian Wisata Berdasarkan Kriteria	80
Kode Sumber 4.16. Implementasi Java Pencarian Cepat dengan Rekomendasi	82
Kode Sumber 4.17. Implementasi Java Detail Wisata	83
Kode Sumber 4.18. Implementasi Java Penunjuk Jalan menggunakan Google Maps.....	83

Kode Sumber 4.19. Implementasi Java Pencarian Terdekat	85
Kode Sumber 4.20. Implementasi <i>Controller</i> Melihat Wisata	87
Kode Sumber 4.21. Implementasi <i>Controller</i> Tambah Wisata Manual	89
Kode Sumber 4.22. Implementasi fungsi <i>Scraping</i>	91
Kode Sumber 4.23. Implementasi <i>Controller</i> Tambah Wisata <i>Scraping Website</i>	93
Kode Sumber 4.25. Implementasi <i>Controller</i> Pengaturan Kota ..	95
Kode Sumber 4.26. Implementasi <i>Controller</i> Pengaturan Kategori	98
Kode Sumber 4.27. Implementasi <i>Controller</i> Pengaturan Wisa	101

BAB I

PENDAHULUAN

Bagian ini akan dijelaskan hal-hal yang menjadi latar belakang, permasalahan yang dihadapi, batasan masalah, tujuan dan manfaat, metodologi, dan sistematika penulisan yang digunakan dalam pembuatan Tugas Akhir aplikasi perangkat bergerak pencarian wisata berdasarkan daerah berbasis android.

1.1. Latar Belakang

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh. Pada perkembangannya sistem operasi Android digunakan pada perangkat *smartphone*. Pada tahun 2014, Android merajai sistem operasi dunia dengan menguasai pasar sebesar 81% jauh meninggalkan para pesaingnya. Strategy Analytics mencatat sistem operasi Android mengapalkan 1 miliar unit *smartphone* [1].

Melihat Perkembangan *smartphone* dengan sistem operasi Android yang begitu pesat, membuat peluang untuk membuat sebuah aplikasi wisata daerah di Provinsi Jawa Timur. Pariwisata di provinsi Jawa Timur memiliki pesona yang sangat unik, sehingga sangat memikat untuk dikunjungi. Provinsi ini sangat kaya akan berbagai tempat alternatif liburan, semuanya tersebar di seluruh daerah kabupaten/kota, setiap kabupaten/kota memiliki ciri khas wisata masing-masing [2]. Namun, kekayaan wisata daerah di provinsi Jawa Timur belum diketahui oleh masyarakat banyak, kebanyakan masyarakat hanya mengetahui wisata yang sudah populer, padahal wisata objek wisata yang lain tidak kalah bagus. Kurangnya pengetahuan tentang wisata daerah di Provinsi Jawa Timur akibat kurangnya penyedia informasi.

Akibat kurangnya penyedia informasi membuat wisata di daerah tempat tinggalnya sendiripun terkadang belum diketahui. Sehingga perlu adanya sistem informasi wisata yang menunjukan wisata-wisata yang terdekat dengan lokasi pengguna. Wisata terdekat terkadang memiliki keindahan yang tidak kalah bagus dan

menarik dengan wisata yang jauh di kota lain yang lebih populer. Jika potensi wisata daerah terdekat diketahui oleh masyarakat daerah tersebut, bila ingin berwisata tidak perlu ke tempat yang jauh/di kota lain dan bisa menghemat pengeluaran.

Setelah mendapat informasi tentang wisata daerah sekitar, kebanyakan masyarakat terkadang tidak mengetahui jalan menuju lokasi tersebut. Maka dari itu, perlu adanya aplikasi yang dapat memberikan penunjuk jalan menuju lokasi wisata dari lokasi pengguna saat ini. Fungsi penunjuk jalan menuju lokasi wisata akan sangat membantu apabila yang akan berwisata adalah orang luar daerah.

1.2. Tujuan

Tujuan dari Tugas Akhir ini adalah sebagai berikut.

1. Terbentuknya aplikasi agar dapat memberikan informasi kepada masyarakat tentang wisata di daerah.
2. Terbentuknya aplikasi yang dapat membantu pihak-pihak terkait dalam mempromosikan wisata yang dimiliki oleh daerah.
3. Terbentuknya aplikasi agar dapat membantu masyarakat dalam melakukan pencarian wisata.

1.3. Rumusan Permasalahan

Rumusan masalah yang diangkat dalam Tugas Akhir ini adalah sebagai berikut.

1. Bagaimana cara mendapatkan data wisata dengan cara *scraping website* wisata lain.
2. Bagaimana cara mengimplementasikan metode *Analytic Hierarchy Proses* (AHP) untuk memberikan rekomendasi wisata kepada pengguna.
3. Bagaimana cara mengimplementasikan rumus Haversine untuk mendapatkan wisata terdekat dengan lokasi pengguna saat ini.
4. Bagaimana cara memberikan penunjuk jalan kepada pengguna aplikasi, dari lokasi saat ini ke lokasi wisata.

1.4. Batasan Permasalahan

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, diantaranya sebagai berikut.

1. Data didapatkan dari internet yang dimasukkan secara *scraping* dan manual dari *website* wisata yang lain.
2. Implementasi metode *Analytic Hierarchy Process* (AHP) hanya mengenai aspek kategori wisata dan aspek biaya masuk wisata.
3. Aplikasi berjalan pada sistem operasi Android minimal versi 4.2.2 (Jelly Bean).
4. Perangkat bergerak yang digunakan minimal memiliki layar 4 inci.
5. Studi kasus tempat wisata yang diambil dari daerah Provinsi Jawa Timur.
6. Aplikasi untuk pengguna merupakan aplikasi perangkat bergerak yang harus dipasang pada perangkat pengguna dan terhubung dengan internet.
7. Aplikasi untuk administrator dibangun berbasis *website* dengan menggunakan Framework Yii.

1.5. Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini yakni sebagai berikut.

1. Studi literatur

Mengumpulkan literatur yang dibutuhkan dalam proses perancangan dan implementasi sistem yang akan dibangun. Sistem yang dimaksud termasuk aplikasi situs web dan perangkat bergerak. Literatur yang dibutuhkan antara lain sebagai berikut.

1. Penggunaan kerangka kerja berbasis *Yii Framework*
2. Proses komunikasi data antara aplikasi berbasis *mobile* dengan aplikasi *website*.
3. Penggunaan metode *Analytic Hierarchy Proses* (AHP) untuk menentukan rekomendasi tempat wisata.

4. Penggunaan metode Haversine untuk menemukan daftar rekomendasi wisata terdekat.

2. Analisis dan Perancangan Sistem

Melakukan analisa kebutuhan sistem sebagai solusi atas permasalahan yang dihadapi pengguna. Dari proses tersebut selanjutnya dirumuskan rancangan sistem yang akan dibangun dan dapat menangani permasalahan. Langkah pada tahap ini antara lain:

- a. Analisa aktor yang terlibat dalam sistem.
- b. Perancangan *use case diagram* sebagai analisa kebutuhan fungsional sistem.
- c. Analisa kebutuhan non-fungsional.
- d. Analisa arsitektur sistem.
- e. Analisa arsitektur perangkat lunak.
- f. Analisa aktifitas kebutuhan fungsional.
- g. Rancangan antarmuka aplikasi *website*.
- h. Rancangan antarmuka perangkat bergerak.

3. Implementasi

Pada tahap ini dilakukan pembuatan elemen perangkat lunak. Sistem yang dibuat berpedoman pada rancangan yang telah dibuat pada proses perancangan dan analisis sistem. Perincian tahap ini adalah sebagai berikut.

- a. Implementasi rancangan aplikasi situs *website* untuk mengelola sistem aplikasi perangkat bergerak pencarian wisata berdasarkan daerah berbasis android.
- b. Implementasi *web service* berfungsi untuk menangani komunikasi aplikasi perangkat bergerak pencarian wisata berdasarkan daerah berbasis android dengan basis data.
- c. Implementasi rancangan aplikasi perangkat bergerak.
- d. Implementasi rancangan basis data.
- e. Implementasi metode menampilkan hasil pencarian.

4. Pengujian dan evaluasi

Pada tahap ini dilakukan pengujian terhadap perangkat lunak berdasarkan skenario yang telah ditentukan. Pengujian ini bertujuan untuk menguji kebutuhan fungsional yang dibutuhkan, masalah yang timbul, kekurangan program dan kemudahan pengguna dalam menggunakan aplikasi. Tahap pengujian yang akan dilakukan sebagai berikut.

- a. Uji coba aplikasi situs *website server* dengan mengelola wisata, mengelola kategori, mengelola kota, dan melihat wisata.
- b. Uji coba aplikasi perangkat bergerak dengan melakukan pencarian sesuai kriteria, pencarian cepat dengan rekomendasi, pencarian terdekat, melihat detail wisata dan menggunakan penunjuk jalan.

5. Penyusunan buku Tugas Akhir

Pada tahap ini dilakukan pendokumentasian dan pelaporan dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan, dan hasil-hasil yang telah didapatkan selama pengerjaan Tugas Akhir.

1.6. Sistematika Penulisan

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini.

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan Tugas Akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

Bab II Dasar Teori

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan Tugas Akhir ini.

Bab III Analisis dan Perancangan Sistem

Bab ini membahas mengenai perancangan perangkat lunak. Perancangan perangkat lunak meliputi perancangan data, arsitektur, proses dan perancangan antarmuka.

Bab IV Implementasi

Bab ini berisi implementasi dari perancangan perangkat lunak.

Bab V Pengujian dan Evaluasi

Bab ini membahas pengujian dengan metode pengujian subjektif untuk mengetahui penilaian aspek kegunaan (*usability*) dari perangkat lunak dan pengujian hasil analisis kakas.

Bab VI Kesimpulan

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan Tugas Akhir.

Lampiran

Merupakan tambahan yang berisi daftar istilah yang penting pada aplikasi ini.

BAB II DASAR TEORI

Pada bab ini dibahas mengenai teori-teori yang menjadi dasar dari pembuatan Tugas Akhir.

2.1. Aplikasi wisata yang sudah ada.

Aplikasi android tentang informasi wisata sudah sangat banyak tersedia di toko aplikasi Play Store. Aplikasi tersebut banyak menyediakan informasi wisata tentang wisata Indonesia hingga wisata luar negeri. Untuk aplikasi wisata di Indonesia kebanyakan menyediakan informasi wisata yang sudah populer, padahal wisata yang belum populer tidak kalah menarik untuk dikunjungi. Dari aplikasi- aplikasi yang sudah ada tersebut, juga tidak ditemukannya rekomendasi wisata dalam fitur-fiturnya, padahal rekomendasi wisata sangat diperlukan oleh masyarakat untuk mendapatkan alternatif liburan. Ada beberapa contoh aplikasi wisata yang tersedia di toko Play Store, diantaranya aplikasi **Wisataku** dan **WisataLokal**. Aplikasi wisataku memiliki keunggulan fitur pencarian dan fitur penunjuk jalan, sedangkan aplikasi WisataLokal memiliki fitur pencarian. Dilihat dari kedua aplikasi tersebut, bisa dikatakan keduanya tidak memiliki fitur rekomendasi wisata, padahal bagi beberapa masyarakat, rekomendasi wisata merupakan sesuatu yang sangat penting untuk memilih alternatif tempat liburan. Dibeberapa aplikasi wisata yang sudah ada, sudah memiliki fitur penunjuk jalan. Fitur penunjuk jalan merupakan fitur yang sangat penting bagi pengguna aplikasi yang tidak mengetahui jalan menuju lokasi wisata. Pengguna yang belum pernah mengunjungi lokasi wisata akan sangat terbantu dengan adanya fitur tersebut.

2.2. Pariwisata Jawa Timur

Provinsi Jawa Timur memiliki pesona wisata yang sangat unik sehingga sangat memikat untuk dikunjungi. Jawa Timur sangat kaya akan berbagai alternatif liburan. Daya tarik atau

tujuan wisata diantaranya : wisata alam, wisata religi, wisata pantai, wisata gunung, wisata pendidikan, wisata kuliner, wisata agro, wisata sejarah, wisata budaya, wisata seni, wisata kerajinan dan masih banyak yang lainnya. Semua wisata tersebar di seluruh daerah kabupaten/kota. Setiap kabupaten/kota memiliki ciri khas wisata masing-masing.

Kota Malang banyak memiliki tempat wisata yang murah dan terkenal, diantaranya Alun-alun kota, Candi Singosari dan Taman Balai Kota. Kota Batu memiliki kegiatan wisata petik apel dan tempat rekreasi keluarga, seperti Jatim Park 1, BNS dan Selecta. Kawah Ijen dapat dikunjungi dari beberapa daerah yakni Kabupaten Banyuwangi, Kabupaten Bondowoso dan Kabupaten Jember. Berwisata bahari di Lamongan dan Telaga Sarangan di Magetan, serta masih banyak tempat wisata lain yang membuat Jawa Timur begitu mengesankan sebagai tujuan liburan dan petulangan. Bagi pecinta wisata belanja, ada objek wisata Intako, yakni tempat industri wisata tas dan koper dan kerajinan dari kulit khas Tanggulangin di Sidoarjo [3].

Untuk menikmati Taman Nasional Bromo-Semeru dapat dilakukan dari beberapa daerah yakni Kabupaten Pasuruan, Kabupaten Malang, Kabupaten Probolinggo dan Kabupaten Lumajang. Taman Nasional Bromo-Semeru merupakan icon wisata Jawa Timur yang dihuni oleh suku Tengger dan setiap tahun diselenggarakan upacara kasodo. Bila berkunjung anda bisa menikmati dinginnya udara yang bersih dan segar berselimutkan awan yang indah. Taman Nasional Bromo-Semeru merupakan satu-satunya kawasan konservasi di Indonesia yang memiliki keunikan berupa lautan pasir seluas 5.250 hektar. Sebuah pulau di bagian timur Jawa yaitu pulau Madura, layak untuk dikunjungi yang terkenal dengan budaya karapan sapi (lomba pacu sapi) [2].

Surabaya dikenal sebagai kota pahlawan, hal ini terjadi sejak pertempuran rakyat Surabaya melawan penjajah Belanda dalam revolusi kemerdekaan Indonesia. Di Surabaya banyak tempat-tempat wisata sejarah, diantaranya Monumen Tugu Pahlawan, Monumen Bambu Runcing dan masih banyak lagi.

Sejak Kota Surabaya dipimpin oleh ibu risma, banyak taman-taman dibuat untuk memberikan wisata tempat liburan masyarakat di Surabaya. Taman Bungkul merupakan taman yang paling terkenal, ta man tersebut memiliki banyak fasilitas, mulai dari oleh raga, tempat kuliner, wifi yang suka berinternet ria [3].

Kayanya objek dan daya tarik wisata daerah di provinsi Jawa Timur masih kurang diketahui oleh masyarakat. Ketidaktahuan masyarakat tentang wisata daerah di Jawa Timur akibat kurang tersedianya informasi. Masyarakat hanya tahu tentang wisata yang sudah populer dan menganggap objek wisata di provinsi Jawa Timur hanya sedikit. Penyedia informasi kebanyakan mengulas wisata-wisata yang sudah terkenal.

2.3. JavaScript Object Notation

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan. JSON mudah untuk dibaca dan ditulis oleh manusia dan memudahkan mesin untuk mengurai dan menciptakan. Format ini berdasar dari himpunan bagian bahasa pemrograman JavaScript standar ECMA-262 edisi ke-3, Desember 1999 [4].

JSON merupakan format teks yang benar – benar berbeda atau independen namun tetap menggunakan konvensi yang mudah dikenali bagi pemrogram yang sudah terbiasa dengan bahasa C, C++, C#, Java, JavaScript, Perl, Python dan bahasa lainnya. Sifat JSON yang seperti ini menjadikannya format pertukaran data yang ideal.

Penggunaan JSON umumnya untuk format transfer data antar perangkat/sistem operasi/bahasa pemrograman yang berbeda dan sebagai media penyimpanan. Contoh penggunaan JSON adalah pembuatan aplikasi yang bersifat *clien-server*, misalnya aplikasi sistem informasi berita, pengelola berita sebagai *server* yang biasanya berupa aplikasi *website* dan pembaca berita sebagai *client* biasanya aplikasi perangkat bergerak. Dalam

pembuatan tugas akhir ini, JSON juga digunakan sebagai penghubung *client* dan *server*.

2.2.1. Protokol Web (SOAP dan REST)

SOAP dan REST adalah protokol untuk berkomunikasi dengan penyedia layanan. SOAP (service-oriented access protocol) adalah protokol komunikasi standar untuk pertukaran pesan berbasis XML. SOAP menggunakan dua protokol *transport*, yakni HTTP dan SMTP. SOAP menggunakan WSDL (*web service description language*) yang berisi kumpulan aturan untuk mendefinisikan pesan, pengikat, operasi, dan lokasi dari *web service* [5].

REST (*representational state transfer*) adalah kumpulan prinsip arsitektur yang mengatur data yang akan ditransmisikan melalui antarmuka HTTP. Pada protokol REST klien dapat mengakses sumber data menggunakan URI. REST mendukung beberapa standar operasi HTTP yakni GET, PUT, DELETE, POST, dan HEAD.

Penggunaan protokol web umumnya untuk aplikasi *client-server*, dimana aplikasi pengguna(*client*) digunakan oleh banyak pengguna sehingga data yang ada pada basis data mungkin digunakan secara bersama-sama dalam satu waktu dan aplikasi administrator(*server*) sebagai pengelolanya. Contoh aplikasi yang menggunakan adalah sistem informasi berita.

2.2.2. MySQL

MySQL merupakan sistem manajemen basis data yang bersifat *open source*. MySQL dikembangkan, didistribusikan dan didukung oleh perusahaan Oracle. MySQL merupakan sistem manajemen basis data yang berfungsi untuk menambah, simpan, dan akses data yang tersimpan di komputer. Basis data MySQL bersifat relasional. Basis data relasional menyimpan data di tabel yang berbeda, tidak menyimpan seluruhnya dalam satu simpanan besar.

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basis data sejak lama, yaitu SQL (*Structured Query*

Language). SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

MySQL layak digunakan karena memiliki banyak keunggulan, diantaranya: bersifat relasional yang artinya menyimpan data pada tabel berbeda, portabilitas artinya dapat berjalan di berbagai sistem operasi, terbuka artinya didistribusikan sebagai perangkat lunak sumber terbuka, *multi-user* artinya dapat digunakan oleh beberapa pengguna secara bersamaan, *performance tuning* yaitu memiliki kecepatan untuk dalam melakukan perintah *query*, *konektivitas* yaitu dapat melakukan koneksi dengan berbagai protokol, keamanan yaitu memiliki beberapa level keamanan, dan masih banyak lagi [6].

2.1.3. Website Scraping

Website scraping adalah aplikasi yang memproses kode HTML dari halaman situs *website* untuk dilakukan ekstraksi data. Ekstraksi data dilakukan untuk memanipulasi data pada situs *website* sumber agar dapat disajikan dalam bentuk lain. Skrip dan aplikasi *website scraping* bekerja dengan mensimulasikan situs *website* ketika ditampilkan melalui peramban. Skrip *website scraping* melakukan koneksi dan pemesanan halaman situs web. Situs *website* sumber kemudian mengirim balik halaman yang dipesan untuk kemudian dilakukan ekstraksi informasi tertentu. *Web scraping* umumnya diterapkan pada situs web tertentu dan mengekstrak informasi tertentu yang berguna misal untuk perbandingan harga. Penggunaan *scraping* dilakukan ketika penyedia konten tidak menyediakan API untuk memudahkan pengembang *mashup* mendapatkan informasi yang mereka inginkan. Proses *scraping* melakukan analisis dan penguraian konten yang sebenarnya didesain untuk konsumsi manusia.

Web crawling adalah sebuah program otomatisasi sederhana atau skrip yang secara runtut memindai halaman internet untuk membentuk indeks data pencarian. Pada umumnya

web crawler hanya digunakan sekali, akan tetapi dapat juga digunakan dalam jangka panjang. Penggunaan jangka panjang contohnya digunakan oleh mesin pencari untuk menyediakan penjelajah situs web dengan situs web yang relevan [7].

Umumnya penggunaan *website scraping* untuk membuat aplikasi perbandingan dengan tujuan bisa membandingkan dua sumber atau lebih dan kemudian ditampilkan dalam satu halaman. Contoh aplikasi yang menggunakan *website scraping* adalah *website* perbandingan harga dan spesifikasi *handphone* (situshp.com, versus.com, gsmarena.com dan lain-lain).

2.4. Analytic Hierarchy Process (AHP)

Analytic Hierarchy Process (AHP) adalah metode yang bisa digunakan untuk membuat keputusan dengan berbagai kriteria dan menyusunnya menjadi sebuah hirarki. Perinsip kerja AHP adalah menyederhanakan masalah kompleks yang tidak terstruktur, strategik dan dinamik menjadi bagian-bagiannya, serta menata *variable* dalam suatu tingkatan. Kemudian tingkat kepentingan *variable* diberi nilai numerik secara subjektif tentang arti pentingnya secara relatif dibandingkan dengan *variabel* lain. Dari berbagai pertimbangan tersebut kemudian dilakukan *sintesa* untuk menetapkan *variabel* yang memiliki prioritas tertinggi dan berperan untuk mempengaruhi hasil pada sistem tersebut [8].

Metode AHP sering digunakan sebagai metode pemecahan masalah dibanding dengan metode yang lain karena memiliki kelebihan sebagai berikut:

1. Struktur yang hirarki, sebagai konsekuensi dari kriteria yang dipilih, sampai pada subkriteria yang paling dalam.
2. Memperhitungkan validitas sampai dengan batas toleransi inkonsistensi berbagai kriteria dan alternatif yang dipilih oleh pengambilan keputusan.
3. Menghitung daya tahan output analisis sensitivitas pengambilan keputusan.

2.5. Metode Haversine

Metode Haversine adalah metode yang digunakan untuk menghitung jarak antar kedua titik di permukaan bumi. Menggunakan garis lintang (longitude) dan garis bujur (latitude) sebagai inputan. Haversine formula adalah persamaan penting pada navigasi memberikan jarak lingkaran besar antar dua titik pada permukaan bola (bumi) berdasarkan bujur dan lintang. Penggunaan rumus ini mengasumsikan pengabaian efek *ellipsoidal*, cukup akurat untuk sebagian besar perhitungan, juga pengabaian ketinggian bukit dan kedalaman lembah di permukaan bumi [5].

Dengan mengasumsikan bumi berbentuk bulat sempurna dengan jari-jari R 6.367, 45 km dan lokasi dari dua titik di koordinat bola (lintang dan bujur) masing-masing adalah $long_1$, lat_1 dan $long_2$, lat_2 maka rumus Haversine dapat ditulis sebagai berikut. Rumus Haversine untuk menghitung jarak terdekat dapat dilihat pada persamaan (2.1), persamaan (2.2), persamaan (2.3), persamaan (2.4) dan persamaan (2.5)

$$\Delta lat = lat_2 - lat_1 \quad (2.1)$$

$$\Delta long = long_2 - long_1 \quad (2.2)$$

$$a = \sin^2[\Delta lat/2] + \cos[lat_1] \cdot \cos[lat_2] \cdot \sin^2[\Delta long/2] \quad (2.3)$$

$$c = 2 \cdot \text{atan2}(\sqrt{a} - \sqrt{(1-a)}) \quad (2.4)$$

$$d = R \cdot c \quad (2.5)$$

2.6. Sistem Operasi Android

Sistem operasi Android merupakan sebuah sistem operasi berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. Sistem operasi Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan diberbagai macam perangkat lunak.

Sistem operasi Android merupakan sistem operasi yang dimiliki oleh Google Inc. Google membeli sistem operasi Android dari perusahaan Android Inc. Pada pengembangan sistem

operasi Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan piranti keras, piranti lunak dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile dan Nvidia. Pada saat perilisan, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan standart terbuka pada perangkat seluler. Di pihak lain Google merilis kode-kode Android dengan lisensi Apache, sebuah lisensi perangkat lunak dan standart terbuka perangkat seluler.

Di dunia terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Services* (GMS) dan tanpa dukungan langsung oleh Google atau benar-benar bebas yang dikenal dengan *Open Handset Distribution* (OHD) [9].

2.7. Android Global Positioning System (GPS)

GPS (*global positioning system*) adalah sistem untuk penentuan lokasi di permukaan bumi dengan bantuan penyelarasan sinyal satelit. Sinyal diterima oleh alat penerima dan digunakan untuk penentuan letak, kecepatan, arah, dan waktu. GPS menggunakan beberapa satelit yang ada pada orbit bumi untuk memancarkan sinyal ke alat penerima. Pada sistem GPS terdapat 3 bagian penting, yakni bagian pengontrol, bagian angkasa, dan bagian pengguna. Bagian pengontrol bertugas untuk melacak orbit, lokasi, ketinggian dan kecepatan satelit. Bagian angkasa terdiri dari satelit-satelit pada orbit bumi. Bagian pengguna adalah alat navigasi yang bertugas menerima data dari satelit.

Android GPS adalah sistem GPS yang terdapat pada perangkat android. Sistem ini dioperasikan oleh Google Location Services API. Google Location Services API adalah layanan penentuan lokasi yang dapat menyederhanakan dan memudahkan pekerjaan seperti penentuan penyedia lokasi dan manajemen daya. Kita dapat menggunakan layanan penentuan lokasi ini untuk membangun aplikasi berbasis lokasi. Layanan penentuan

lokasi ini menggunakan dua macam penyedia untuk menentukan lokasi pengguna, yakni penyedia lokasi GPS dan penyedia lokasi jaringan. Layanan ini memungkinkan aplikasi melakukan tiga hal. Fungsi pertama melakukan pencarian terhadap daftar penyedia lokasi. Fungsi kedua melakukan perintah untuk pemutakhiran lokasi pengguna secara berkala. Fungsi ketiga adalah melakukan perintah untuk memicu dijalankannya intent, jika lokasi pengguna memenuhi syarat lokasi yang telah ditentukan [5].

Penyedia lokasi GPS memberikan lokasi yang terakurat dibanding penyedia lokasi lainnya. Kelemahan dari penyedia lokasi ini adalah hanya bekerja pada ruang terbuka, menguras baterai dan membutuhkan waktu yang lama untuk melakukan pencarian. Penyedia lokasi jaringan menentukan lokasi pengguna dengan menggunakan menara jaringan dan sinyal wifi. Penyedia lokasi ini dapat memberikan lokasi baik ketika pengguna berada di ruang terbuka maupun ruang tertutup, dapat memberikan respon lokasi dengan cepat, serta tidak menguras baterai. Alur untuk mendapatkan lokasi pengguna adalah sebagai berikut.

1. Memulai aplikasi dan mulai mendengarkan pemutakhiran dari penyedia lokasi yang diinginkan.
2. Mendapatkan estimasi terbaik dari lokasi saat ini.
3. Berhenti mendengarkan pemutakhiran lokasi.
4. Menggunakan estimasi lokasi yang telah didapatkan.

2.8. Google Maps

Google Maps adalah layanan peta digital paling populer dan terefektif yang ada di internet saat ini. Google Maps menyediakan gambar pemetaan dari seluruh permukaan bumi. Layanan ini juga dapat menyediakan penunjuk arah ke setiap lokasi tujuan dan bahkan menghitung jarak terpendek dan rute perjalanan termurah.

Google Maps yang digunakan pada aplikasi android menggunakan layanan Google Maps Android API v2. Layanan ini dapat memunculkan penanda untuk mengidentifikasi suatu

tempat, memunculkan objek di atas peta, dan menempel satu atau lebih peta sebagai fragment.

Dengan Google Maps Android API v2, kita dapat menempelkan peta pada aplikasi yang sedang dibangun ke dalam activity sebagai fragment dengan kode XML yang sederhana. Fitur yang terdapat pada Google Maps API adalah peta tiga dimensi, peta dalam ruangan, satelit, tanah, dan campuran. Fitur lain yang dimiliki oleh Google Maps adalah efek animasi pada transisi.

Google Maps Android API juga menyertakan fitur penanda pada peta. Penanda dapat ditambahkan pada peta untuk menandai tempat khusus bagi pengguna. Selain penanda, Google Maps juga dapat memungkinkan aplikasi menggambarkan polylines dan poligons untuk menandai rute dan wilayah pada peta. Google Maps API memungkinkan pengguna untuk mendapatkan berbagai perspektif pada peta dengan fungsi rotasi dan pembesaran [5].

2.9. Google Places

Google Places adalah salah satu layanan Google yang menggunakan basis data yang sama dengan Google Maps dan Google+ Local. Google Places API memungkinkan aplikasi berbasis lokasi mendapatkan informasi detail mengenai tempat dari berbagai kategori. Google Places API dilengkapi dengan lebih dari 95 juta lokasi bisnis dan lokasi menarik yang diperbarui secara berkala. Pembaruan dapat dilakukan melalui pemilik bisnis yang terdaftar maupun kontribusi dari pengguna.

Google Places API memungkinkan pengguna untuk melakukan pencarian terhadap detail informasi dari suatu tempat dengan kategori yang bermacam-macam seperti perusahaan, lokasi menarik, lokasi geografis, dan sebagainya. Google Places API juga dapat terintegrasi kedalam Google Maps API sebagai Pustaka Javascript. Google Places API juga menyertakan dukungan terhadap kejadian (semua jenis acara perkumpulan, penampilan, dan promosi baik privat maupun umum). Dengan

dukungan terhadap kejadian, Google Places memperbolehkan kita untuk mencari dan menampilkan kejadian yang digelar pada lokasi yang terdaftar pada Google Places [5].

Google Places dapat membantu pengguna melakukan pengambilan keputusan mengenai lokasi yang akan dituju, mulai dari restoran, hotel, hingga toko. Layanan ini juga dapat memberikan informasi lokasi non-bisnis seperti museum, sekolah, dan taman.

[Halaman ini sengaja dikosongkan]

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tahap analisis permasalahan dan perancangan dari sistem yang dibangun. Analisis permasalahan membahas permasalahan yang diangkat dalam pengerjaan Tugas Akhir. Analisis kebutuhan mencantumkan kebutuhan-kebutuhan yang diperlukan perangkat lunak. Selanjutnya dibahas mengenai perancangan sistem yang dibuat. Pendekatan yang dibuat dalam perancangan ini adalah pendekatan berorientasi objek. Perancangan direpresentasikan dengan diagram Unified Modelling Language (UML).

3.1. Analisis

Tahap analisis dibagi menjadi beberapa bagian antara lain cakupan permasalahan, deskripsi umum sistem, kasus penggunaan sistem, dan kebutuhan perangkat lunak.

3.1.1. Analisis Permasalahan

Pariwisata Jawa Timur sangat kaya akan berbagai macam objek dan daya tarik wisata [2]. Semua objek dan daya tarik wisata tersebut tersebar di berbagai daerah kabupaten/kota. Setiap kabupaten/kota memiliki ciri khas wisata masing-masing. Sayangnya objek wisata di provinsi Jawa Timur kurang diketahui oleh masyarakat. Umumnya masyarakat hanya mengetahui wisata yang sudah populer. Ketidaktahuan masyarakat akibat kurangnya informasi yang disediakan.

Banyak *website* dan aplikasi perangkat bergerak yang menyediakan informasi wisata di Jawa Timur, misalnya untuk *website* seperti www.eastjava.com, www.indonesia.travel dan www.wisata-jatim.com, sedangkan aplikasi wisata bergerak aplikasi Wisataku, WisataLokal dan tripadvisor. Penyedia informasi *website* dan aplikasi perangkat bergerak tersebut kebanyakan mengulas wisata yang sudah populer di masyarakat. Tujuan dari *website* dan aplikasi perangkat bergerak tersebut

memang untuk mendapatkan rating tinggi dari masyarakat, tetapi akibat dari itu, banyak wisata yang lain belum terinformasikan kepada masyarakat. Wisata yang belum terinformasikan ke masyarakat terkadang memiliki keindahan yang tidak kalah bagus dengan wisata populer. Pantai Coro di Tulungagung adalah salah satu pantai terindah di kota tersebut. Untuk menuju pantai tersebut, pengunjung tidak dapat menggunakan mobil atau motor, pengunjung harus jalan kaki sejauh 5 Km. Walaupun perjalanan jauh, pengunjung tidak akan menyesalnya karena pantai ini memiliki keindahan lebih dari pantai yang lainnya.

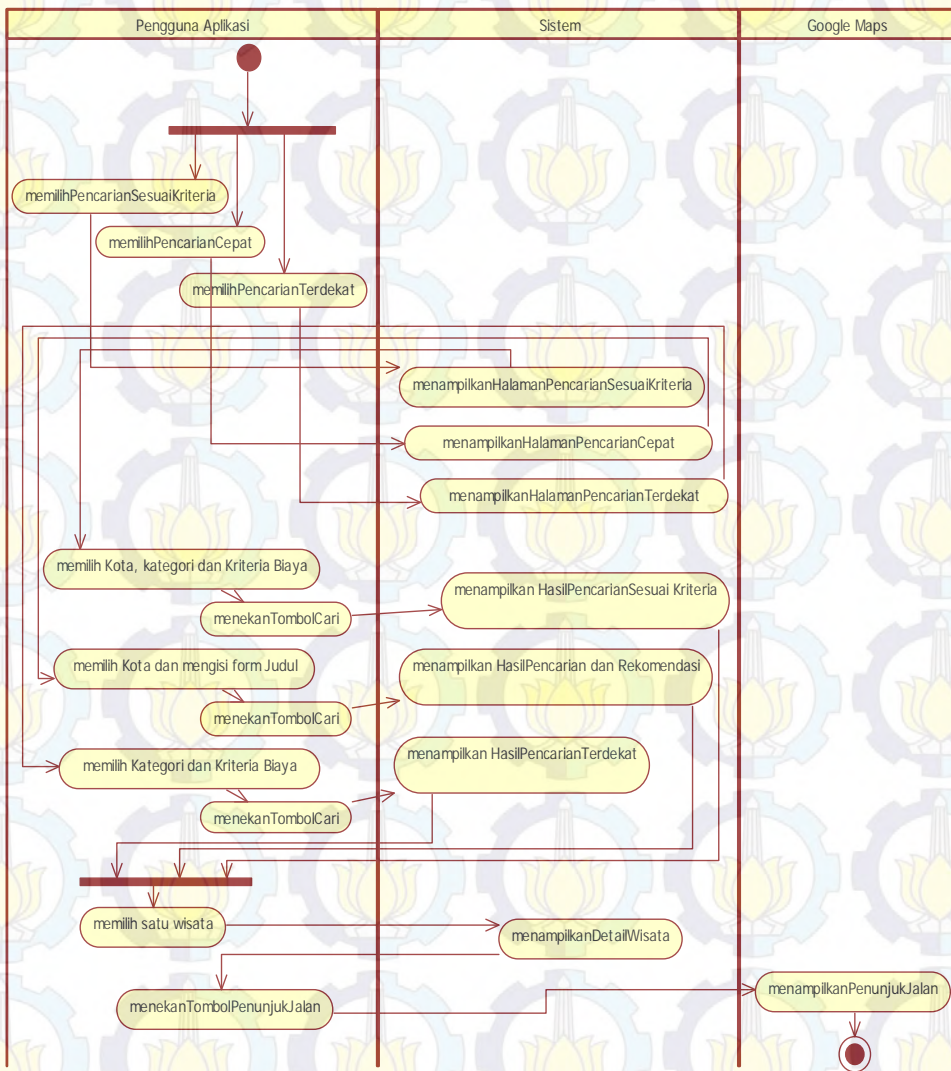
Tidak terinformasikannya tentang objek wisata yang belum populer di provinsi Jawa Timur ke masyarakat, membuka peluang untuk membuat sebuah aplikasi pencarian wisata daerah yang menyediakan informasi wisata secara keseluruhan setiap daerah. Aplikasi tersebut dapat digabungkan dengan *website* wisata yang sudah ada dengan cara *scraping website* dan menggabungkan dengan data yang ditambahkan secara langsung oleh pengelola aplikasi.

Untuk memberikan informasi ke masyarakat tentang wisata yang belum populer, diperlukan rekomendasi wisata agar bila pengguna melakukan pencarian wisata populer, maka rekomendasi memunculkan wisata yang hampir sama kriterianya dengan wisata yang dicari. Rekomendasi wisata tersebut akan membuat wisata yang belum terkenal menjadi wisata alternatif. Metode *Analytic Hierarchy Proses* (AHP) sangat cocok untuk digunakan memberikan rekomendasi wisata.

Dalam berwisata, masyarakat tidak langsung memutuskan wisata yang ingin dikunjungi. Masyarakat juga akan melihat waktu yang tersedia dan jarak yang akan ditempuh. Jika berwisata dengan jarak yang jauh, maka secara otomatis diperlukan waktu yang cukup banyak. Dari permasalahan tersebut dibutuhkan aplikasi yang dapat memberikan rekomendasi wisata terdekat dengan lokasi pengguna sekarang ini. Rumus Haversine sangat cocok untuk digunakan memberikan rekomendasi wisata terdekat.

Dalam permasalahan berikutnya, pengguna setelah mendapatkan informasi tentang wisata alternatif dari rekomendasi yang didapatkan, biasanya pengguna tidak mengetahui akan lokasi dan jalan menuju ke lokasi wisata tersebut, sehingga diperlukan sebuah informasi penunjuk jalan yang akan menuntun pengguna ke lokasi wisata.

Untuk menjawab semua permasalahan di atas, dalam Tugas Akhir ini dibuat sebuah aplikasi wisata (WisataINA) yang memiliki fungsionalitas sesuai dengan permasalahan. Aplikasi WisataINA memiliki fitur pencarian wisata berdasarkan kriteria, fitur pencarian wisata dengan rekomendasi, fitur pencarian wisata terdekat dengan lokasi pengguna saat ini dan fitur penunjuk jalan ke lokasi wisata. Berikut Alur proses pencarian wisata dapat dilihat pada gambar 3.1.



Gambar 3.1. Alur proses pencarian aplikasi perangkat bergerak

Data aplikasi WisataINA didapatkan dengan menambahkan informasi wisata secara manual dan *scraping website* wisata yang lain. Informasi yang diambil antara lain adalah judul, ulasan, gambar, dan alamat website. Semua informasi tempat wisata yang ditambahkan secara manual dan *scraping website* wisata lain disimpan ke dalam satu basis data. Untuk melakukan pencarian wisata bagi pengguna, perangkat harus terhubung ke internet. Aplikasi melakukan pencarian pada basis data *online* yang telah disiapkan, jika daftar wisata ditemukan, aplikasi akan menampilkannya ke pengguna. Pengguna dapat memilihi detail wisata dengan memilih satu wisata. Detail wisata berisi judul, ulasan, gambar, dan alamat *website*. Pengguna juga dapat menggunakan fitur penunjuk jalan untuk ke lokasi wisata.

3.1.1.1. Analisis permasalahan mendapatkan data dengan *scraping website*

Pada permasalahan ini sistem dituntut dapat mendapatkan data secara *scraping* dari *website* wisata lain.

Website scraping adalah proses untuk mendapatkan data dari *website* lain dengan cara ekstraksi. Ekstraksi data dilakukan untuk mengambil data konten pada situs *website* wisata sumber agar dapat disajikan dalam bentuk lain. *Website scraping* mengambil data dari *website* sumber ketika ditampilkan melalui peramban. *Scraping website* umumnya diterapkan pada situs *website* tertentu dengan tujuan tertentu. Penggunaan *scraping* dilakukan ketika penyedia konten tidak menyediakan API untuk memudahkan pengembang mendapatkan informasi yang mereka inginkan. Pada kasus ini, untuk mendapatkan data wisata dengan *scraping website*, admin harus mengetahui lokasi *script* HTML *website* sumber yang menyediakan konten judul wisata, ulasan wisata, gambar wisata. Konten yang akan dilakukan *scraping* harus memiliki *script* yang unik atau beda dengan konten lain pada *website* sumber, sehingga bila dilakukan *scraping* sistem tidak melakukan kesalahan.

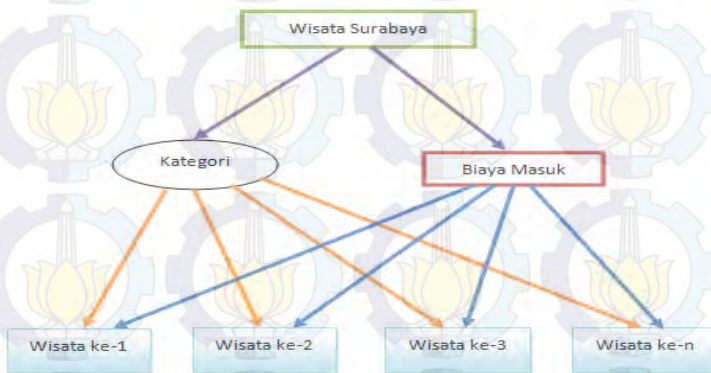
3.1.1.2. Analisis permasalahan penerapan metode *Analytic Hierarchy Process* (AHP)

Pada permasalahan ini sistem dituntut untuk bisa memberikan rekomendasi wisata kepada pengguna. Rekomendasi wisata didapatkan dari perhitungan dengan metode *Analytic Hierarchy Process* (AHP), kemudian sistem menampilkan wisata yang dicari dan rekomendasi wisata yang memiliki kriteria hampir sama dengan wisata yang dicari pengguna. Langkah-langkah untuk mendapatkan rekomendasi wisata dengan menggunakan AHP dibahas pada sub bab ini.

Untuk mendapatkan rekomendasi wisata dengan menggunakan algoritma AHP ada prinsip dasar yang harus dilakukan yaitu sebagai berikut:

a. Membuat Hirarki (*Decomposition*)

Adalah memecahkan atau membagi permasalahan yang utuh menjadi unsur-unsur ke bentuk hirarki proses pengambilan keputusan, dimana setiap unsur saling berhubungan. Struktur hirarki untuk permasalahan pencarian tempat wisata dapat dilihat pada Gambar 3.2, keterangan kategori wisata dapat dilihat pada Tabel 3.1, dan keterangan kriteria biaya masuk dapat dilihat pada Tabel 3.2.



Gambar 3.2. Hirarki permasalahan dalam algoritma AHP

Keterangan :

Tabel 3.1. Kriteria Kategori

Kategori	1. Alam
	2. Religi
	3. Sejarah
	4. Budaya
	5. Belanja
	6. Kuliner

Tabel 3.2. Kriteria Biaya Masuk

Kategori Biaya	Range(Rp.)
Murah	Rp 0 – Rp 50.000
Sedang	Rp 50.001 – Rp 150.000
Mahal	Diatas Rp 150.000

Nilai Kriteria biaya masuk didapatkan dari hasil implementasi survei kriteria biaya masuk (Terlampir pada Lampiran B).

b. Penilaian Kriteria (*Comparative Judgement*)

Penilaian yang dilakukan berdasarkan kepentingan relatif dua elemen pada satu tingkat tertentu dalam kaitannya dengan tingkat di atasnya. Proses ini merupakan inti dalam metode AHP karena berpengaruh terhadap urutan prioritas dari elemennya. Skala yang digunakan yaitu skala Saaty yang menyatakan intensitas kepentingan. Skala Saaty dapat dilihat pada tabel 3.3.

Tabel 3.3. Skala Saaty

Nilai	Deskripsi
1	Kedua kriteria sama penting
3	Kriteria yang satu sedikit lebih penting dari pada kriteria yang lainnya.
5	Kriteria yang satu lebih penting dari pada yang lainnya.
2,4	Nilai tengah-tengah

c. Menentukan Prioritas (*Synthesis of Priority*).

Adalah proses menentukan prioritas dari elemen kriteria, dilakukan berdasarkan pandangan para pakar dan pihak-pihak yang berkepentingan dengan menggunakan *eigen vektor* untuk mendapatkan bobot alternatif untuk rekomendasi atau pengambilan keputusan.

d. Konsistensi Logis (*Logical Consistency*)

Adalah mengagresikan seluruh *eigen vektor* yang diperoleh dari tingkatan hirarki dan selanjutnya diperoleh suatu vektor komposit yang menghasilkan urutan rekomendasi atau pengambilan keputusan.

Setelah perinsip dasar dilakukan, selanjutnya melakukan perhitungan untuk mendapatkan bobot kriteria dan mendapatkan rekomendasi wisata. Langkah-langkah perhitungannya adalah sebagai berikut ini.

Langkah ke-1.

Membuat Matrik Perbandingan berpasangan dengan menggunakan skala Saaty (Tabel 3.3) sebagai nilainya, nilai prioritas kriteria didapat dari survei (terlampir pada lampiran C). Matrik Perbandingan dapat dilihat pada Tabel 3.4 dan Matrik Perbandingan Penyederhanaan dapat dilihat pada Tabel 3.5.

Tabel 3.4. Matrik Perbandingan

Matrik Perbandingan		
	Biaya Masuk	Kategori
Biaya Masuk	1	1/3
Kategori	3	1

Tabel 3.5. Matrik Perbandingan Penyederhanaan

Matrik Perbandingan(Penyederhanaan)		
	Biaya Masuk	Kategori
Biaya Masuk	1	0.33
Kategori	3	1
Σ Kolom	4	1.33

Langkah ke-2.

Menghitung bobot kriteria (priority vector) dengan cara :

- 1) Normalisasi nilai setiap kolom matrik perbandingan berpasangan dengan membagi setiap nilai pada kolom matrik dengan hasil penjumlahan kolom yang bersesuaian.
- 2) Menghitung nilai rata-rata dari penjumlahan setiap baris matrik. Hasil perhitungan matrik yang dinormalisasi dapat dilihat pada Tabel 3.6.

Tabel 3.6. Matrik Perbandingan Normalisasi.

Matrik Perbandingan(Normalisasi)				
	Biaya Masuk	Kategori	Jumlah Baris	Eigen Vector
Biaya Masuk	0.25	0.25	0.5	0.25
Kategori	0.75	0.75	1.5	0.75

Berikut adalah perhitungan bobot relatif yang dinormalkan.

$$1 : 4 = 0,25 \quad 0,33 : 1,33 = 0,25$$

$$3 : 4 = 0,75 \quad 1 : 1,33 = 0,75$$

Berikut adalah perhitungan Eigen Vector :

$$\text{Eigen Vektor Biaya Masuk} = \Sigma \text{ Baris/Kolom} = 0.5/2 = 0,25$$

$$\text{Eigen Vektor Kategori} = \Sigma \text{ Baris/Kolom} = 1.5/2 = 0,75$$

$$\lambda_{\max} = (0,25*4) + (0,75*1.33) = 1,9975$$

Langkah ke-3.

Mengecek konsistensi *Ratio* (CR) dari matrik perbandingan berpasangan kriteria. Jika $CR > 0.1$ maka harus diulang kembali perbandingan berpasangan sampai didapat $CR \leq 0.1$.

Consistency index, CI ditemukan

$$CI = (\lambda_{\max} - n) / (n-1) = (1,9975-2) / (2-1) = -0,0025$$

Nilai RI (Inkonsistensi Rasio) matrik dapat dilihat pada Tabel 3.7

Tabel 3.7. Tabel Inkonsistensi Rasio

Ordo Matrik	1	2	3	4
Nilai RI	0.00	0.00	0.58	0.90

Sehingga $CR = CI/CR = -0,0025/0.00 = -$ tak terhitung ($CR \leq 0.1$, Maka konsisten)

Langka ke-4.

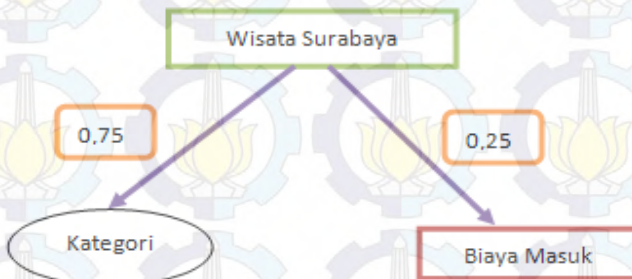
Menyusun Tree dengan bobot.

Karena $CR \leq 0,1$ atau konsisten, Dapat diperoleh nilai

Biaya masuk : $0,25 \times 100\% = 25\%$

Kategori : $0,75 \times 100\% = 75\%$

Tree yang telah memiliki bobot dapat dilihat pada Gambar 3.3.



Gambar 3.3. Tree yang memiliki Bobot

Pada Gambar 3.3 menunjukan bahwa kriteria kategori lebih penting dari pada kriteria biaya masuk. Hal ini ditunjukan dengan nilai bobot kategori sebesar 75% dan biaya masuk sebesar 25%.

3.1.1.3. Analisis Permasalahan Penerapan Metode *Haversine*

Metode *Haversine* adalah formula untuk menghitung jarak terdekat kedua titik. Di permukaan bumi menggunakan garis lintang dan garis bujur. Persamaan *Haversine* menghitung permukaan bumi dengan membuat lingkaran antar dua titik berdasarkan garis lintang dan garis bujur.

Pada permasalahan ini, Rumus *Haversine* digunakan untuk memberikan rekomendasi wisata terdekat dengan lokasi pengguna sekarang ini. Dalam implementasi Metode *Haversine*, digunakan lokasi pengguna sebagai sebagai pusat lingkaran. Untuk titik kedua, merupakan titik secara acak dengan jarak maksimal yang ditentukan oleh pembuat aplikasi, sehingga dengan Metode *Haversine* dapat dibuat sebuah lingkaran antar kedua titik tersebut. Wisata dengan lokasi di dalam lingkaran yang terbentuk dengan Metode *Haversine* merupakan rekomendasi wisata. Setelah mendapatkan daftar wisata rekomendasi, dilakukan perhitungan untuk mendapatkan wisata terdekat sampai wisata terjauh dari lokasi pengguna tapi masih di dalam radius lingkaran yang terbentuk.

3.1.1.4. Analisis permasalahan penunjuk jalan ke lokasi wisata dengan Google Maps

Pada permasalahan ini, aplikasi yang dibuat dituntut untuk dapat memberikan informasi penunjuk jalan dari lokasi pengguna ke lokasi wisata.

Pada ponsel pintar dengan sistem operasi Android saat ini, sudah disediakan perangkat keras yang digunakan untuk mengunci posisi ponsel pintar digunakan (GPS). Posisi ponsel pintar digunakan merupakan posisi pengguna aplikasi WisataINA yang dapat berubah-ubah. Pada ponsel pintar juga terdapat aplikasi peta dunia yang merupakan bawaan dari sistem operasi Android. Aplikasi peta dunia tersebut adalah Google Maps. Aplikasi Google Maps memiliki fitur penunjuk jalan dimana tingkat keakuratannya hampir 100%.

Dengan adanya aplikasi Google Maps dengan tingkat keakuratan hampir 100% pada ponsel pintar dengan sistem operasi Android, pengembang aplikasi pencarian wisata (WisataINA) mencoba untuk memanfaatkannya sebagai sebuah fitur yang dapat membantu pengguna aplikasi dalam mencari jalan menuju tempat lokasi wisata. Aplikasi WisataINA akan mendeteksi lokasi pengguna menghubungkannya dengan koordinat lokasi wisata yang telah tersimpan pada basis data.

3.1.2. Deskripsi Umum Sistem

Aplikasi yang dibuat adalah aplikasi informasi wisata dengan konsep *client-server*. Aplikasi *server* adalah aplikasi berbasis *website* yang digunakan oleh administrator untuk mengelola data wisata.

Aplikasi *client* adalah aplikasi pengguna untuk melakukan pencarian wisata berdasarkan daerah. Aplikasi ini berjalan pada ponsel pintar dengan sistem operasi Android. Aplikasi pencarian wisata memiliki spesifikasi dari mulai tingkat kota. Aplikasi dibagi menjadi beberapa kategori wisata, yaitu alam, religi, belanja, sejarah, budaya, dan kuliner. Sedangkan dalam kriteria biaya dibagi beberapa kriteria biaya masuk, yaitu murah (Rp0-Rp50.000), sedang (Rp50.001-Rp150.000) dan mahal (lebih dari Rp150.000). Aplikasi memiliki beberapa fitur, yaitu fitur pencarian sesuai kriteria, pencarian dengan rekomendasi, pencarian terdekat dan fitur penunjuk jalan. Perangkat lunak ini diharapkan dapat membantu pengguna dalam melakukan pencarian wisata di daerah dan membantu menuju ke lokasi wisata.

3.1.3. Aktor

Aktor mendefinisikan entitas-entitas yang terlibat dan berinteraksi langsung dengan sistem. Entitas ini bisa berupa manusia maupun sistem atau perangkat lunak yang lain. Terdapat dua aktor yang dapat menggunakan sistem ini yaitu sebagai berikut.

a. Pengguna aplikasi

Pengguna aplikasi adalah pengguna yang menggunakan aplikasi ini untuk mencari wisata pada daerah tertentu.

b. Administrator situs *website*

Administrator adalah pengguna yang menggunakan aplikasi *website* untuk mengelola data (tambah, edit dan delete).

3.1.4. Kebutuhan Fungsional

Kebutuhan fungsional berisi proses-proses yang harus dimiliki sistem. Kebutuhan fungsional mendefinisikan layanan yang harus disediakan dan reaksi sistem terhadap masukan atau pada situasi tertentu. Daftar kebutuhan fungsional dapat dilihat pada Tabel 3.8.

Tabel 3.8. Daftar Kebutuhan Fungsional Perangkat Lunak

Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi
F-0001	Mencari Tempat Wisata	Pengguna dapat mencari wisata sesuai kriteria pengguna.
F-0002	Melihat Detail	Pengguna dapat melihat detail wisata dengan memilih satu wisata yang ingin dilihat
F-0003	Menggunakan penunjuk jalan	Pengguna menggunakan penunjuk jalan ke lokasi wisata
F-0004	Mengelola Kota	Administrator dapat mengelola kota melalui aplikasi <i>website</i> .
F-0005	Mengelola Kategori	Administrator dapat mengelola kategori melalui aplikasi <i>website</i> .
F-0006	Mengelola Wisata	Administrator dapat mengelola wisata melalui aplikasi <i>website</i> .

3.1.5. Kebutuhan Non Fungsional

Kebutuhan non fungsional terdiri dari faktor-faktor selain kebutuhan fungsional. Kebutuhan non fungsional mendefinisikan

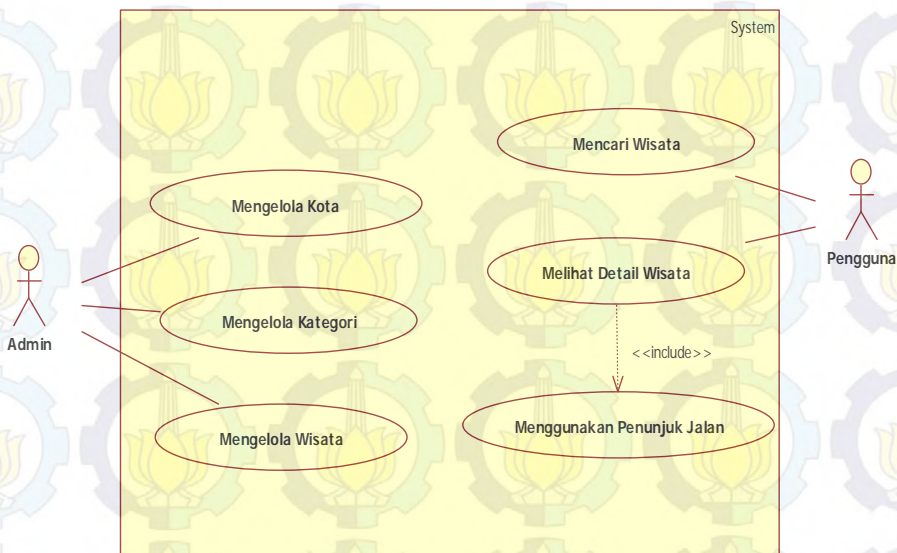
faktor-faktor yang mendukung jalannya sebuah sistem. Daftar kebutuhan non fungsional dapat dilihat pada Tabel 3.9.

Tabel 3.9. Daftar Kebutuhan Non Fungsional Perangkat Lunak

Kode Kebutuhan	Kebutuhan Non Fungsional	Deskripsi
NF-0001	Usability	Pengguna dapat mengoperasikan aplikasi dengan mudah.

3.1.6. Kasus Penggunaan

Berdasarkan analisis spesifikasi kebutuhan fungsional dan analisis aktor dari sistem dibuat kasus penggunaan sistem. Kasus-kasus penggunaan dalam sistem ini dijelaskan secara rinci pada subbab ini. Kasus penggunaan digambarkan dalam sebuah diagram kasus penggunaan. Diagram kasus penggunaan dapat dilihat pada Gambar 3.4, dan penjelasan dari setiap kasus penggunaan dapat dilihat pada Tabel 3.10



Gambar 3.4. Diagram Kasus Penggunaan

Tabel 3.10. Daftar Kode Diagram Kasus Penggunaan

Kode Kasus Penggunaan	Nama
UC-0001	Mengelola Kota
UC-0002	Mengelola Kategori
UC-0003	Mengelola Wisata
UC-0004	Mencari Wisata
UC-0005	Melihat detail
UC-0006	Menggunakan penunjuk jalan

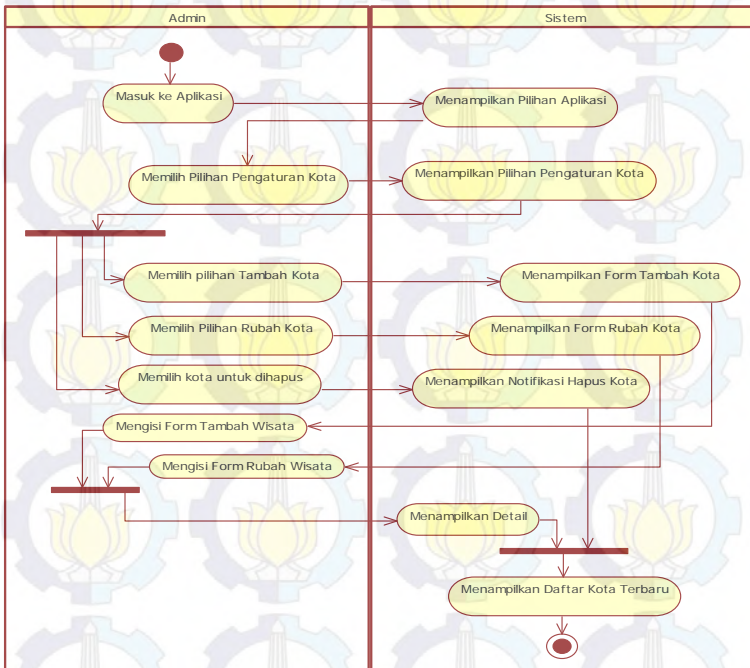
3.1.6.1. Mengelola Kota

Pada kasus penggunaan ini, Administrator dapat mengelola (menambah, menghapus dan merubah) data kota. Spesifikasi kasus penggunaan ini dapat dilihat pada. Spesifikasi kasus mengelola kota dapat dilihat pada Tabel 3.11. Diagram aktivitas mengelola kota dapat dilihat pada Gambar 3.5.

Tabel 3.11. Spesifikasi Kasus Mengelola Kota

Nama	Mengelola Kota
Kode	UC-0001
Deskripsi	Mengelola data kota
Tipe	Fungsional
Pemicu	Administrator memilih menu pengaturan Kotadan kemudian masuk ke halaman kota.
Aktor	Administrator
Kondisi Awal	Administrator sudah melakukan login pada web mnajemen pencarian wisata.
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Administrator memilih menu pengaturan Kota. 2. Sistem menampilkan halaman pengaturan Kota yang isinya daftar kota. 3. Administrator memilih sub menu lihat Kota. 4. Sistem menampilkan detail kota berupa idkota, idprovinsi dan namakota.
	<ol style="list-style-type: none"> 1. Administrator memilih menu pengaturan

- Kejadian Alternatif 1	<p>kota.</p> <ol style="list-style-type: none">2. Sistem menampilkan halaman pengaturan kota yang isinya daftar kota.3. Administrator memilih sub menu ubah kota.4. Sistem menampilkan form ubah kota.5. Pengguna mengubah nama kota.6. Pengguna menekan tombol simpan.7. Sistem menyimpan data perubahan dan menampilkan detail hasil perubahan.
-Kejadian Alternatif 2	<ol style="list-style-type: none">1. Administrator memilih menu pengaturan kota.2. Sistem menampilkan halaman pengaturan kota yang isinya daftar kota.3. Administrator memilih sub menu hapus kota.4. Sistem menampilkan konfirmasi penghapusan.5. Sistem menampilkan daftar kota terbaru.
Kondisi Akhir	Sistem menampilkan daftar kota terbaru.



Gambar 3.5. Diagram Aktivitas Mengelola Kota

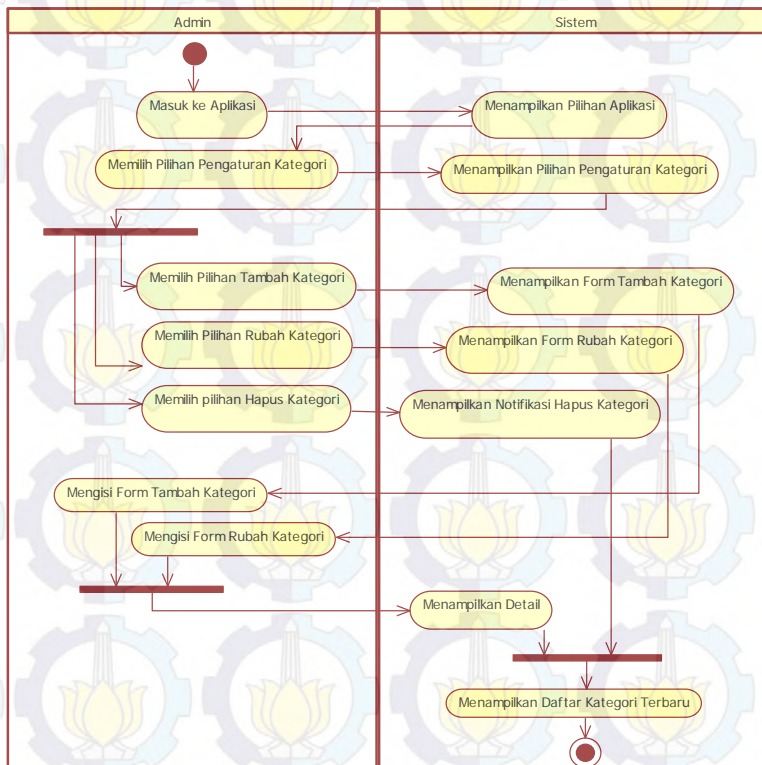
3.1.6.2. Mengelola Kategori

Pada kasus penggunaan ini, Administrator dapat mengelola (menambah, menghapus dan merubah) data kategori. Spesifikasi kasus penggunaan ini dapat dilihat pada. Spesifikasi kasus mengelola kategori dapat dilihat pada Tabel 3.12. Diagram aktivitas mengelola kategori dapat dilihat pada Gambar 3.6.

Tabel 3.12. Spesifikasi Kasus Mengelola Kategori

Nama	Mengelola Kategori
Kode	UC-0002
Deskripsi	Mengelola data kategori
Tipe	Fungsional
Pemicu	Administrator memilih menu pengaturan

	kategori dan kemudian masuk ke halaman kategori.
Aktor	Administrator
Kondisi Awal	Administrator sudah melakukan login pada web manajemen pencarian wisata.
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Administrator memilih menu pengaturan kategori. 2. Sistem menampilkan halaman pengaturan kota yang isinya daftar kategori. 3. Administrator memilih sub menu lihat kategori. 4. Sistem menampilkan detail kota berupa id kategori dan nama kategori.
- Kejadian Alternatif 1	<ol style="list-style-type: none"> 1. Administrator memilih menu pengaturan kategori. 2. Sistem menampilkan halaman pengaturan kategori yang isinya daftar kategori. 3. Administrator memilih sub menu ubah kategori. 4. Sistem menampilkan form ubah kategori. 5. Pengguna mengubah nama kategori. 6. Pengguna menekan tombol simpan. 7. Sistem menyimpan data perubahan dan menampilkan detail hasil perubahan.
- Kejadian Alternatif 2	<ol style="list-style-type: none"> 1. Administrator memilih menu pengaturan kategori. 2. Sistem menampilkan halaman pengaturan kategori yang isinya daftar kategori. 3. Administrator memilih sub menu hapus kategori. 4. Sistem menampilkan konfirmasi penghapusan. 5. Sistem menampilkan daftar Kategori terbaru.
Kondisi Akhir	Sistem menampilkan daftar Kategori terbaru.



Gambar 3.6. Diagram Aktivitas Mengelola Kategori

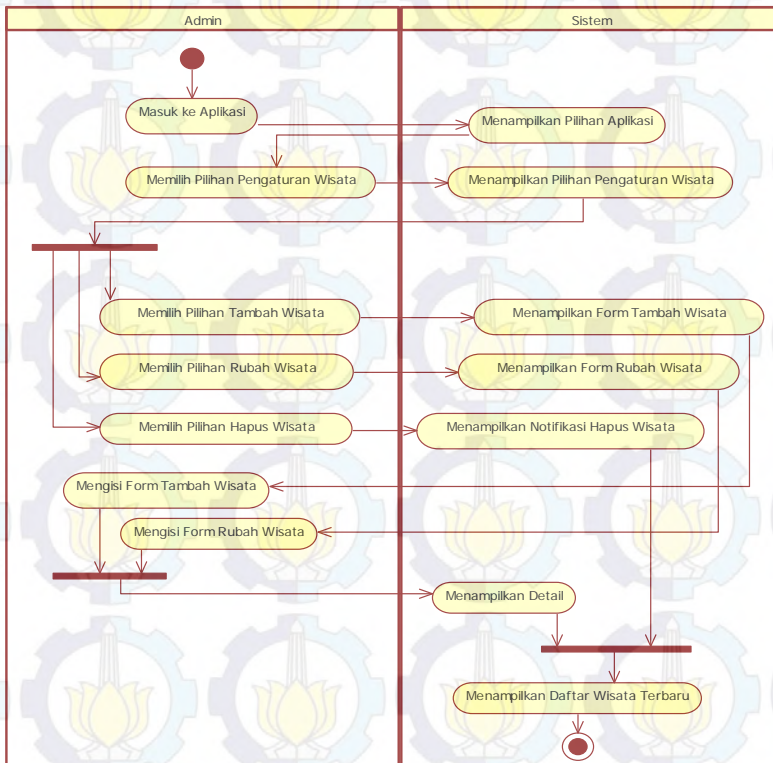
3.1.6.3. Mengelola Wisata

Pada kasus penggunaan ini, Administrator dapat mengelola (menambah, menghapus dan merubah) data Tempat Wisata. Spesifikasi kasus mengelola wisata dapat dilihat pada Tabel 3.13. Diagram aktivitas mengelola wisata dapat dilihat pada Gambar 3.7.

Tabel 3.13. Spesifikasi Kasus Mengelola Tempat Wisata

Nama	Mengelola Wisata
Kode	UC-0003
Deskripsi	Mengelola wisata

Tipe	Fungsional
Pemicu	Administrator memilih menu pengaturan wisata.
Aktor	Administrator
Kondisi Awal	Administrator sudah melakukan login pada web manajemen pencarian wisata.
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Administrator memilih menu pengaturan wisata. 2. Sistem menampilkan halaman pengaturan tempat wisata yang isinya daftar tempat wisata. 3. Administrator memilih submenu lihat tempat wisata 4. Sistem menampilkan detail tempat wisata berupa idtempatwisata, idkategori, judul, ulasan, alamat sumber, hotel, dan foto.
- Kejadian Alternatif 1 -Kejadian Alternatif 2	<ol style="list-style-type: none"> 1. Administrator memilih menu pengaturan tempat wisata. 2. Sistem menampilkan halaman pengaturan tempat wisata yang isinya daftar tempat wisata. 3. Administrator memilih submenu ubah tempat wisata. 4. Sistem menampilkan form ubah tempat wisata. 5. Pengguna mengubah nama isi tempat wisata. 6. Pengguna menekan tombol simpan. 7. Sistem menyimpan data perubahan dan menampilkan detail hasil perubahan.
Kondisi Akhir	Sistem menampilkan daftar tempat wisata terbaru.



Gambar 3.7. Diagram Aktivitas Mengelola Wisata

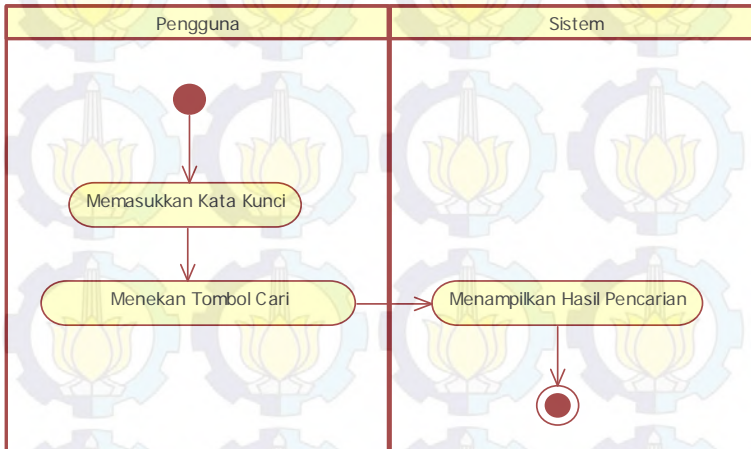
3.1.6.4. Mencari Wisata

Pada kasus penggunaan ini, sistem menerima masukan yang diinginkan oleh pengguna. Sistem menampilkan daftar wisata yang sesuai preferensi dari pengguna. Spesifikasi kasus mencari wisata dapat dilihat pada Tabel 3.14. Diagram aktivitas mencari wisata dapat dilihat pada Gambar 3.8.

Tabel 3.14. Spesifikasi Kasus Mencari Wisata.

Nama	Manajemen Kota
Kode	UC-0004

Deskripsi	Mencari wisata
Tipe	Fungsional
Pemicu	Pengguna membuka halaman pencarian wisata
Aktor	Pengguna aplikasi.
Kondisi Awal	Pengguna sudah masuk kedalam halaman pencarian tempat wisata.
Aliran:	
- Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna memilih menu pencarian wisata. 2. Sistem menampilkan halaman pencarian tempat wisata. 3. Pengguna memilih provinsi, kota, kategori dan memasukkan judul wisata 4. Sistem menampilkan daftar tempat wisata yang sesuai.
- Kejadian Alternatif 1	<ol style="list-style-type: none"> 1. Pengguna memilih menu pencarian tempat wisata. 2. Sistem menampilkan halaman pencarian tempat wisata. 3. Pengguna memilih provinsi, kota dan kategori, tanpa memasukkan judul wisata 4. Sistem menampilkan daftar tempat wisata yang sesuai.
Kondisi Akhir	Sistem menampilkan daftar tempat wisata yang sesuai dengan keinginan pengguna



Gambar 3.8. Diagram Aktivitas Mencari Wisata

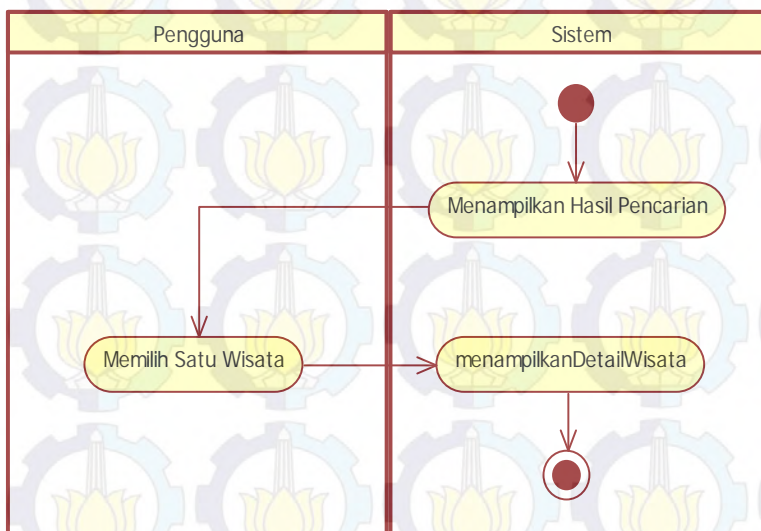
3.1.6.5. Melihat Detail Wisata

Pada kasus penggunaan ini, sistem menerima masukan berupa judul wisata yang telah dipilih oleh pengguna. Sistem menampilkan detail wisata yang dipilih oleh pengguna. Detail wisata berisi judul, ulasan, alamat sumber dan gambar wisata. Spesifikasi kasus melihat detail wisata dapat dilihat pada Tabel 3.15. Diagram aktivitas melihat detail wisata dapat dilihat pada Gambar 3.9.

Tabel 3.15. Spesifikasi Kasus Melihat Detail Wisata.

Nama	Melihat detail wisata
Kode	UC-0005
Deskripsi	Melihat detail wisata.
Tipe	Fungsional
Pemicu	Pengguna memilih satu wisata
Aktor	Pengguna
Kondisi Awal	Pengguna memilih satu wisata dari hasil pencarian

- Aliran :	1. Pengguna memilih satu wisata 2. Sistem menampilkan halaman detail wisata.
- Kejadian Normal	
- Kejadian Alternatif	Tidak ada
Kondisi Akhir	Sistem menampilkan detail wisata
Kebutuhan Khusus	Tidak ada



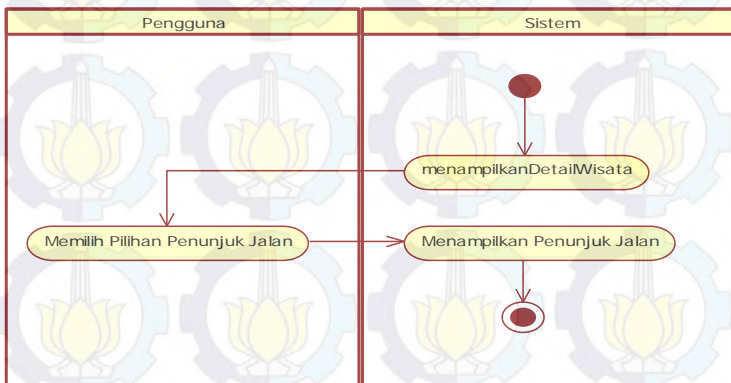
Gambar 3.9. Diagram Aktivitas Melihat Detail Wisata

3.1.6.6. Menggunakan Penunjuk Jalan

Pada kasus penggunaan ini, pengguna dapat melihat lokasi tempat wisata dan penunjuk jalan ke lokasi tempat wisata dengan menggunakan Google Maps. Spesifikasi kasus menggunakan penunjuk jalan dapat dilihat pada Tabel 3.16. Diagram aktivitas menggunakan penunjuk jalan dapat dilihat pada Gambar 3.10.

Tabel 3.16. Spesifikasi Kasus Penggunaan Peta Wisata

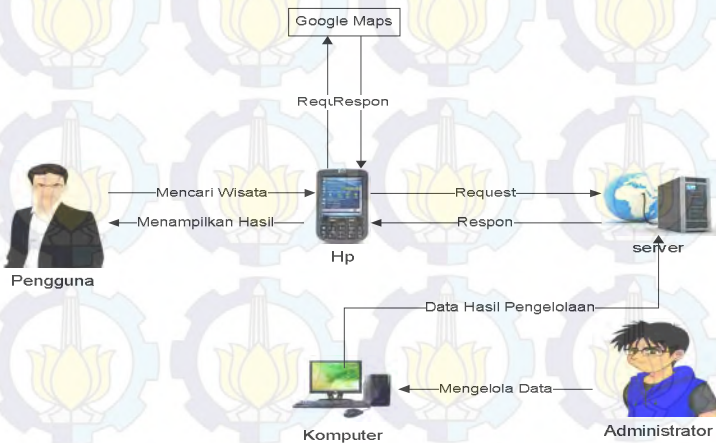
Nama	Memasukkan Data Obat
Kode	UC-0006
Deskripsi	Melihat posisi tempat wisata pada oogle maps.
Tipe	Fungsional
Pemicu	Pengguna melihat detail tempat wisata.
Aktor	Pengguna
Kondisi Awal	Pengguna sudah melihat detail tempat wisata
Aliran:	
- Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna menekan tombol peta. 2. Sistem menampilkan halaman peta lokasi tempat wisata.
- Kejadian Alternatif	Tidak ada
Kondisi Akhir	Sistem menampilkan lokasi tempat wisata.
Kebutuhan Khusus	Tidak ada

**Gambar 3.10. Diagram Aktivitas Penggunaan Penunjuk Jalan.**

3.1.7. Diagram Arsitektur Aplikasi

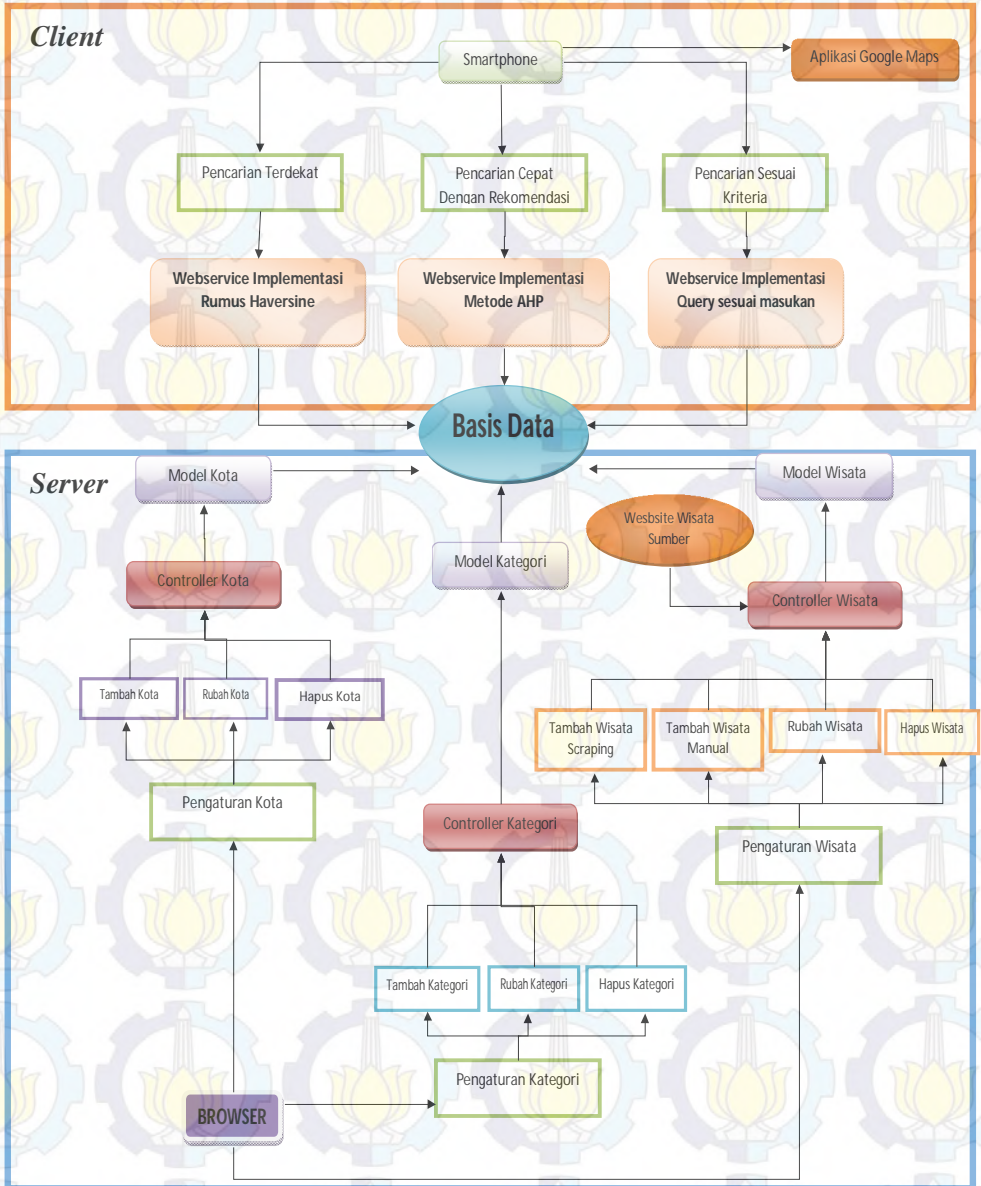
Arsitektur aplikasi ada 2 macam, yaitu arsitektur sistem dan arsitektur perangkat lunak.

Arsitektur sistem adalah pemetaan atau desain kebutuhan-kebutuhan sistem informasi untuk di dalam membangun sebuah aplikasi. Dalam membuat aplikasi sistem informasi wisata (WisataINA) perlu digambarkan arsitektur sistem. Diagram arsitektur dari sistem aplikasi WisataINA digambarkan pada Gambar 3.11.



Gambar 3.11. Diagram Arsitektur Sistem WisataINA

Arsitektur perangkat lunak adalah struktur sistem dari suatu program yang terdiri dari komponen-komponen perangkat lunak untuk membangun sebuah aplikasi. Untuk arsitektur sistem aplikasi informasi wisata (WisataINA) masukan dari pengguna melalui perangkat bergerak dikirim ke *web service*. *Web service* akan melakukan *query* pada basis data. Pada aplikasi *server* masukan dari pengguna melalui peramban akan dikirim oleh *controller* ke model. Model akan melakukan *query* pada basis data. Diagram arsitektur perangkat lunak dapat dilihat pada Gambar 3.12



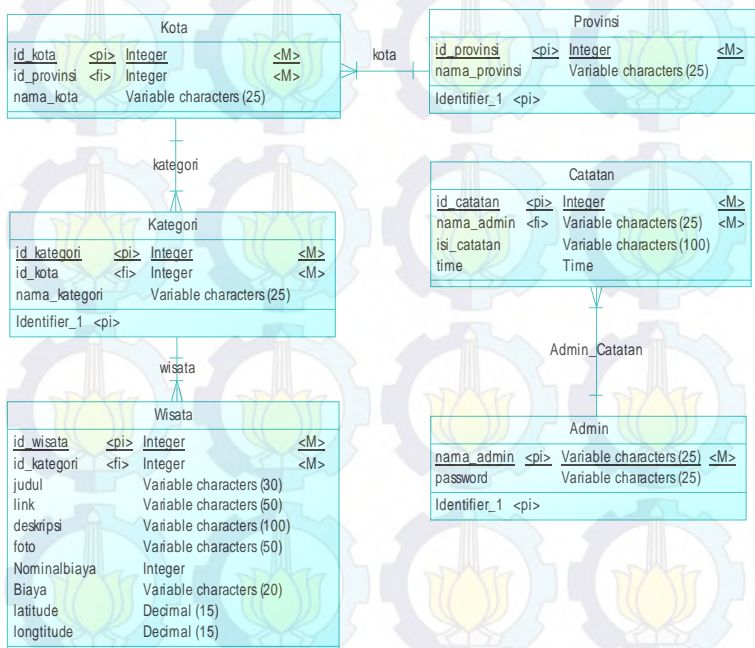
Gambar 3.12. Diagram Arsitektur Perangkat Lunak WisataINA

3.2. Perancangan Basis Data

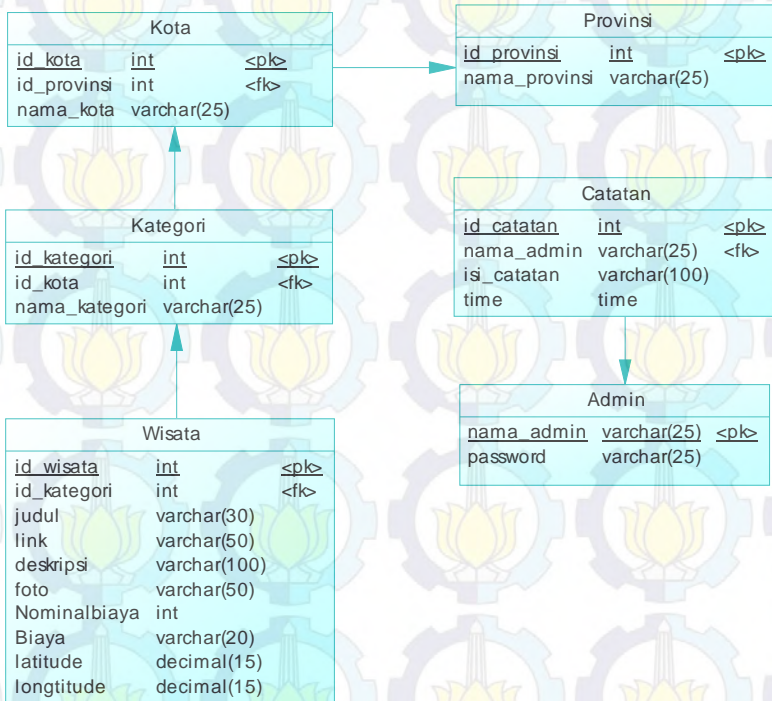
Basis data merupakan sekumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut.

Perangkat lunak yang digunakan untuk mengelola dan memanggil *query* basis data disebut sistem manajemen basis data, atau biasa disebut DBMS. Pada sistem ini, DBMS yang digunakan yakni MySQL (PHPMyAdmin).

Ilustrasi perancangan basis data digambarkan menggunakan CDM (Conceptual Data Model) pada Gambar 3.13 dan PDM (Physical Data Model) seperti pada Gambar 3.14. Spesifikasi basis data sistem dijelaskan pada Tabel 3.17.



Gambar 3.13. Conceptual Data Model Pencarian Wisata



Gambar 3.14. Physical Data Model Basis Data Obat

Tabel 3.17. Spesifikasi Basis Data Pencarian Wisata

No	Tabel	Atribut	Tipe Data	Fungsi
1.	Admin	IDADMIN	int	Menyimpan data administrator
		NAMAADMIN	varchar(100)	
		PASSWORD	char(15)	
2.	Catatan	IDCATATAN	int	Meyimpan data catatan
		IDADMIN	int	
		ISICATATAN	text	
3	Provinsi	IDPROVINSI	int	Menyimpan data provinsi
		NAMAPROVINSI	char(100)	
4	Kota	IDKOTA	int	Menyimpan data kota
		IDPROVINSI	int	
		NAMAKOTA	char(100)	

5	Kategori	IDKATEGORI	int	Menyimpan data kategori
		IDKOTA	int	
		NAMAKATEGORI	char(100)	
6	Tempat Wisata	IDTEMPAWISATA	int	Menyimpan data tempat wisata
		IDKATEGORI	int	
		JUDUL	varchar(100)	
		LINK	varchar(100)	
		DESKRIPSI	varchar(100)	
		FOTO	varchar(100)	
		BIAYA	varchar(100)	
		KOORDINATX	varchar(100)	
		KOORDINATY	varchar(100)	

[Halaman ini sengaja di kosongkan]

BAB IV IMPLEMENTASI

Pada bab ini dijelaskan mengenai implementasi dari perancangan perangkat lunak. Implementasi yang dijelaskan meliputi lingkungan pembangunan perangkat lunak, implementasi antarmuka pengguna, dan implementasi proses.

4.1. Lingkungan Pembangunan Perangkat Lunak

Perangkat keras yang digunakan dalam pembangunan perangkat lunak ini berupa laptop dengan spesifikasi *processor* Intel Core i3-3217U @ 2.40 GHz (4CPU), *memory* 2.99 MB RAM. Perangkat lunak pada laptop yang digunakan dalam pembangunan perangkat lunak antara lain:

- Sistem Operasi Windows 7 Ultimate 32-bit
- StarUML 5.0.2.1570
- Eclipse Platform Version: 4.2.1.v20130118-173121-9MF7GHYdG0B5kx4E_SkfZV-1mNjVATf67ZAb7
- Android Developer Tools Build: v22.3.0-887826
- Notepad++ v6.5.5
- XAMPP Control Panel v3.2.1
- MYSQL Server Version: 5.5.36-cll – MySQL Community Server (GPL)
- phpMyAdmin Version: 4.1.8

4.2. Implementasi Antarmuka Pengguna Admin

Implementasi antarmuka pengguna admin adalah antarmuka yang berfungsi untuk menyajikan berbagai fitur untuk mengelola informasi mulai dari mengelola wisata, mengelola kategori, mengelola kota, dan mengelola provinsi.

4.2.1. Antarmuka Administrator Login

Antarmuka administrator login adalah antarmuka yang berfungsi untuk masuk dalam aplikasi *website* oleh administrator. Administrator bertugas mengelola data pada Aplikasi Pencarian Wisata Berdasarkan Daerah di Android.

Setelah Administrator melakukan *login*, Aplikasi akan menampilkan daftar menu administrator Pencarian, Tambah, Pengaturan Kota, Pengaturan Kategori, Pengaturan Provinsi dan pengaturan Kota. Tampilan antarmuka Administrator Login dapat dilihat pada Gambar 4.1.

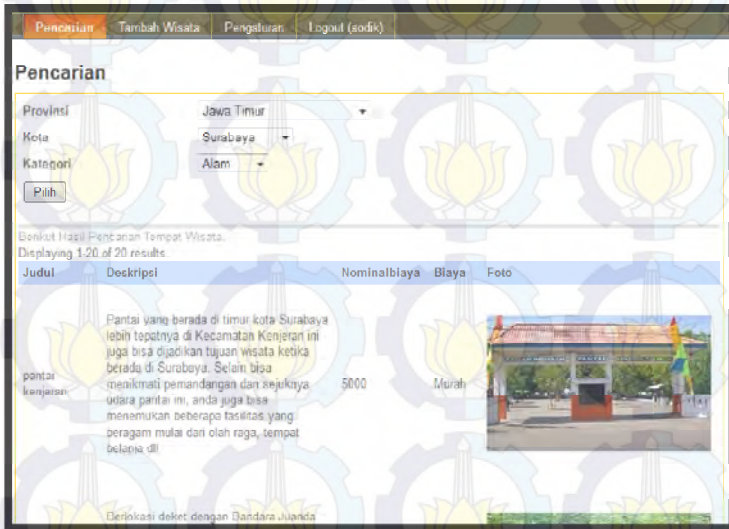


Gambar 4.1. Antarmuka Administrator Login

4.2.2. Antarmuka Adminstrator Melihat Wisata

Antarmuka administrator melihat wisata adalah antar muka yang berfungsi untuk administrator mencari atau melihat data tempat wisata.

Pada antarmuka administrator pencarian, administrator harus memilih provinsi, kota dan kategori yang sesuai dengan keinginan administrator kemudian sistem menampilkan daftar tempat wisata yang sesuai. Tampilan antarmuka Administrator Pencarian dapat dilihat pada Gambar 4.2.



Gambar 4.2. Antarmuka Administrator Pencarian

4.2.3. Antarmuka Administrator Tambah Wisata Manual

Antarmuka Tambah Wisata Manual adalah antarmuka yang berfungsi untuk menambahkan wisata dengan manual. Menambahkan wisata secara manual adalah admin menuliskan judul, ulasan, alamat sumber, foto dan koordinat tempat wisata. Untuk sumber informasi yang ditambahkan oleh administrator berasal dari internet. Tampilan antarmuka tambah wisata manual dapat dilihat pada Gambar 4.3 dan antar muka hasil tambah wisata manual dapat dilihat pada Gambar 4.4.

Tambah Wisata Manual

Kategori dengan * wajib diisi.

Provinsi	Jawa Timur
Kota	Surabaya
Kategori	Kuliner
Judul	Bebek Oven Mozzarella pak joss
Deskripsi	<p>Berawal dari gerobak yang keliling di sekitar perumahan menggunakan jingle. Menu best seller-nya adalah Bebek Oven Mozzarella. Bebek dipotong kecil-kecil, disajikan di hotplate, dioven dan disiram mozzarella di atasnya.</p>
Alamat Web Sumber	http://centaperut.com/user/BebekPakJoss
Foto	Upload maksimal gambar 300px Browse No file selected hapus bebek_pakjoss.png
Biaya	50000 *Biaya masuk/harga perporsi makanan
Tempat	:
Koordinat X	-7.289451
Koordinat Y	112.755617
<input type="button" value="Tambah"/> <input type="button" value="Batal"/>	

Gambar 4.3. Antarmuka Tambah Wisata Manual

Pencarian	Tambah Wisata	Pengaturan	Logout (sodik)
-----------	---------------	------------	----------------

[Home](#) » [Wisata](#) » bebek oven mozzarella pak joss

Detail Wisata bebek oven mozzarella pak joss

Idtempatwisata 459

Judul bebek oven mozzarella pak joss

Link <http://centaperut.com/user/BebekPakJoss>

Deskripsi Berawal dari gerobak yang keliling di sekitar perumahan menggunakan jingle. Menu best seller-nya adalah Bebek Oven Mozzarella. Bebek dipotong kecil-kecil, disajikan di hotplate, dioven dan disiram mozzarella di atasnya.

Foto [bebek_pakjoss.png](#)

Biaya Murah

Nominalbiaya 50000

Koordinatx -7.2894510

Koordinaty 112.7556170

Gambar 4.4. Antarmuka Hasil Tambah Wisata Manual

4.2.4. Antarmuka Administrator Tambah Wisata dengan *Scraping Website* lain.

Antarmuka administrator tambah wisata *scraping* adalah antarmuka yang digunakan oleh administrator untuk menambahkan data wisata dari *website* lain dengan cara *scraping*.

Scraping sendiri adalah mengambil data dari website lain untuk disimpan kedalam basis data dan menampilkannya kedalam *website* yang telah dibuat. Tampilan antarmuka tambah wisata dengan *website scraping* dapat dilihat pada gambar 4.5 dan hasil *website scraping* dapat dilihat pada gambar 4.6.

Pencarian | **Tambah Wisata** | Pengaturan | Logout (sadm)

Home » Tambah Web

Tambah Wisata Website

Kolom dengan * diisi secara manual.

Provinsi	Jawa Timur
Kota	Tulungagung
Kategori	Alam
Alamat Web Sumber	ava.com/tourism/tulungagung/ina/patriok_gebang.html
Judul	<h2> </h2>
Deskripsi	<p> </p>
Foto	http://www.ava.id/ava.com/tourism/tulungagung/ina/
Biaya Masuk	0 *Biaya masuk/tiagra perporisi makanan
Tempat	:
Koordinat X	8.302182
Koordinat Y	111.801330

Tambah Data

Gambar 4.5. Antarmuka Tambah Wisata Dengan *Website Scraping*



Gambar 4.6. Antarmuka Hasil Tambah Wisata Website Scraping

4.2.5. Antarmuka Administrator Mengelola Kota

Antarmuka administrator mengelola kota adalah antarmuka yang berfungsi menampilkan menu pengaturan kota. Administrator dapat menambah, merubah dan menghapus kota sesuai dengan kebutuhan.

Untuk melakukan pengaturan kota, administrator harus memilih menu pengaturan kota pada aplikasi *website*. Administrator dapat menambah kota dengan memilih menu tambah, merubah kota dengan menu rubah dan dapat menghapus kota dengan menu hapus kota. Halaman ini juga dapat melakukan fungsi pencarian. Tampilan antarmuka adminstrator mengelola kota dapat dilihat pada Gambar 4.7.



Gambar 4.7. Antarmuka Adminisrator Mengelola Kota

4.2.6. Antarmuka Administrator Mengelola Kategori

Antarmuka administrator mengelola kategori adalah antarmuka yang berfungsi menampilkan menu pengaturan kategori. Administrator dapat menambah, merubah dan menghapus kategori sesuai dengan kebutuhan.

Untuk melakukan pengaturan kategori, administrator harus memilih menu pengaturan kategori pada aplikasi *website*. Administrator dapat menambah kategori dengan memilih menu tambah, merubah kategori dengan menu rubah dan dapat menghapus kategori dengan menu hapus kategori. Halaman ini juga dapat melakukan fungsi pencarian. Tampilan antarmuka administrator mengelola kategori dapat dilihat pada Gambar 4.8.



Gambar 4.8 Antarmuka Adminstrator Mengelola Kategori

4.2.7. Antarmuka Administrator Mengelola Wisata

Antarmuka administrator mengelola wisata adalah antar muka yang berfungsi menampilkan menu pengaturan wisata. Administrator dapat menambah, merubah dan menghapus wisata sesuai dengan kebutuhan.

Untuk melakukan pengaturan wisata, administrator harus memilih menu pengaturan wisata pada aplikasi *website*. Administrator dapat menambah wisata dengan memilih menu tambah, merubah wisata dengan menu rubah dan dapat menghapus wisata dengan menu hapus wisata. Halaman ini juga dapat melakukan fungsi pencarian. Tampilan antarmuka administrator mengelola wisata dapat dilihat pada Gambar 4.9.



Gambar 4.9 Antarmuka Administrator Mengelola Wisata

4.3. Implementasi Antarmuka Pengguna Aplikasi

Implementasi antarmuka pengguna aplikasi adalah antarmuka yang berfungsi untuk menyajikan berbagai fitur pencarian pada aplikasi android.

4.3.1. Antarmuka Menu Aplikasi

Antarmuka menu aplikasi adalah antarmuka yang berfungsi untuk menampilkan menu aplikasi pengguna pada aplikasi android.

Menu pada aplikasi pengguna ada 2, yaitu menu pencarian dan tentang aplikasi. Antarmuka Menu Aplikasi dapat dilihat pada Gambar 4.10.



Gambar 4.10. Antarmuka Menu Aplikasi

4.3.2. Antarmuka Menu Pencarian

Antarmuka menu pencarian adalah antarmuka yang berfungsi untuk menampilkan menu pencarian wisata dengan metode tertentu.

Menu pencarian pada aplikasi ada 3, yaitu pencarian berdasarkan kriteria, pencarian cepat dengan rekomendasi dan pencarian terdekat. Tampilan antar muka menu pencarian dapat dilihat pada Gambar 4.11.

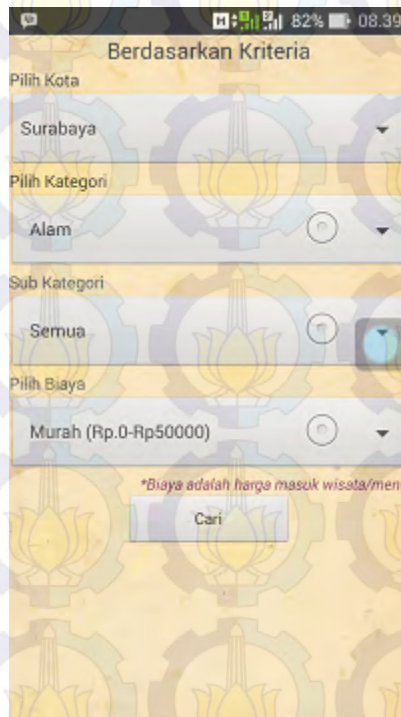


Gambar 4.11. Antarmuka Menu Pencarian.

4.3.3. Antarmuka Pencarian Berdasarkan Kriteria

Antarmuka pencarian berdasarkan kriteria adalah antarmuka yang berfungsi untuk menampilkan halaman kriteria yang harus dipilih oleh pengguna. Pengguna dapat memilih daerah/kota, memilih kategori, dan memilih range biaya yang diinginkan.

Pada menu pencarian ini, pengguna dapat memilih kota yang ada di Jawa Timur, memilih 6 kategori dan memilih 3 pilihan biaya. Tampilan antarmuka pencarian berdasarkan kriteria dapat dilihat pada Gambar 4.12.



Gambar 4.12. Antarmuka Pencarian Berdasarkan Kriteria.

4.3.4. Antarmuka Hasil Pencarian Berdasarkan Kriteria

Antarmuka hasil pencarian berdasarkan kriteria adalah antarmuka yang berfungsi untuk menampilkan hasil pencarian berdasarkan kriteria.

Hasil pencarian ditampilkan pada listview dan pengguna dapat memilih satu wisata untuk melihat detail wisata. Tampilan antarmuka hasil pencarian berdasarkan kriteria dapat dilihat pada Gambar 4.13.



Gambar 4.13. Antarmuka Hasil Pencarian Berdasarkan Kriteria

4.3.5. Antarmuka Pencarian Cepat

Antarmuka pencarian cepat adalah antarmuka yang berfungsi untuk mencari wisata yang diinginkan oleh pengguna dengan cara memasukkan nama tempat wisata. Sebelum memasukkan tempat wisata pengguna diharuskan untuk memilih daerah terlebih dahulu.

Pada menu pencarian ini, aplikasi tidak hanya menampilkan wisata yang telah dicari oleh pengguna, aplikasi juga memberikan rekomendasi wisata lain yang hampir sama kriterianya dengan wisata yang dicari oleh pengguna. Tampilan antarmuka pencarian cepat dapat dilihat pada Gambar 4.14.



Gambar 4.14. Antarmuka Pencarian Cepat

4.3.6. Antarmuka Detail Wisata

Antarmuka detail wisata adalah antarmuka yang berfungsi untuk menampilkan detail wisata yang telah dipilih oleh pengguna.

Detail wisata akan menampilkan judul, alamat sumber website, ulasan tentang wisata dan gambar wisata. Alamat sumber website merupakan sebuah tautan, jika pengguna ingin melihat langsung sumber data dapat mengklik tautan tersebut. Antarmuka detail wisata dapat dilihat pada Gambar 4.15.

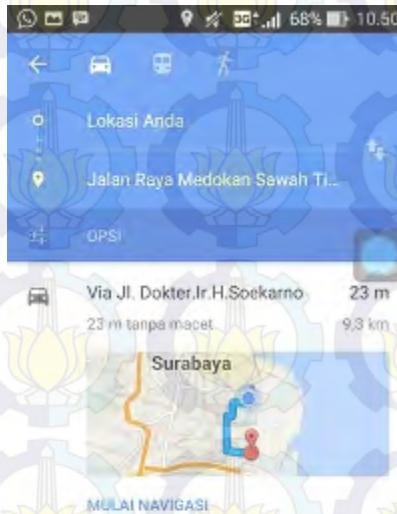


Gambar 4.15. Antarmuka Detail Wisata

4.3.7. Antarmuka Penunjuk Jalan.

Antarmuka penunjuk jalan adalah antarmuka yang berfungsi untuk menampilkan google maps. Google Maps disini digunakan untuk memberikan penunjuk jalan dari posisi pengguna saat ini ke tempat wisata.

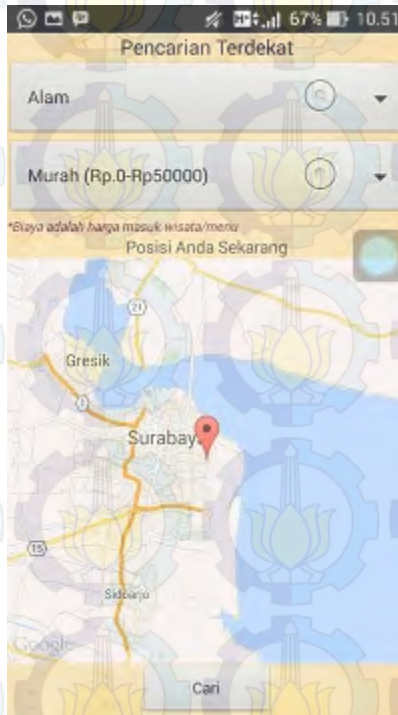
Untuk menggunakan fitur ini, pengguna melihat detail wisata kemudian mengklik tombol penunjuk jalan. Penunjuk jalan akan memberikan pilihan rute yang ingin dilewati oleh pengguna. Tampilan antarmuka penunjuk jalan dapat dilihat pada Gambar 4.16.



Gambar 4.16. Antarmuka Penunjuk Jalan

4.3.8. Antarmuka Pencarian Terdekat

Antarmuka pencarian terdekat adalah antarmuka yang berfungsi untuk mencari wisata yang terdekat dengan lokasi pengguna. Pengguna perlu memilih kategori dan range biaya untuk melakukan pencarian tersebut. Tampilan antarmuka pencarian terdekat dapat dilihat pada Gambar 4.17.



Gambar 4.17 Antarmuka Pencarian Terdekat

4.4. Implementasi Proses

Bagian ini menjelaskan tentang implementasi proses-proses pada perangkat lunak yang dikembangkan dalam Tugas Akhir ini. Penjelasan mengenai implementasi proses pada rancangan bangun aplikasi pengguna dan aplikasi *website* administrator. Implementasi proses aplikasi pengguna meliputi pencarian berdasarkan kriteria, pencarian cepat dengan rekomendasi, dan pencarian terdekat. Sedangkan aplikasi administrator meliputi mengelola provinsi, mengelola kota, mengelola kategori, mengelola wisata serta pengumpulan informasi obat dengan metode *website scraping* dan manual.

4.4.1. Implementasi Basis Data

Implementasi basis data adalah implementasi perintah yang digunakan untuk membuat basis data manajemen basis data MySQL. Terdapat 6 tabel yang digunakan dalam basis data MySQL aplikasi pencarian wisata berdasarkan daerah, yaitu tabel admin, tabel catatan, tabel provinsi, tabel kota, tabel kategori, tabel wisata. Implementasi tabel dapat dilihat pada kode sumber 4.1. Implementasi tabel admin, kode sumber 4.2. Implementasi tabel catatan, kode sumber 4.3. Implementasi tabel provinsi, kode sumber 4.4. Implementasi tabel kota, kode sumber 4.5. Implementasi tabel kategori dan kode sumber 4.6. Implementasi tabel wisata.

```
create table ADMIN
(
  NAMA_ADMIN      varchar(25) not null,
  PASSWORD        varchar(25),
  primary key (NAMA_ADMIN)
);
```

Kode Sumber 4.1. Implementasi Table Admin.

```
create table CATATAN
(
  ID_CATATAN      int not null,
  NAMA_ADMIN      varchar(25) not null,
  ISI_CATATAN     varchar(100),
  TIME            time,
  primary key (ID_CATATAN)
);
```

Kode Sumber 4.2. Implementasi Table Catatan.

```
create table PROVINSI
(
  ID_PROVINSI     int not null,
  NAMA_PROVINSI   varchar(25),
  primary key (ID_PROVINSI)
);
```

Kode Sumber 4.3. Implementasi Tabel Provinsi


```
create table KOTA
(
  ID_KOTA          int not null,
  ID_PROVINSI      int not null,
  NAMA_KOTA        varchar(25),
  primary key (ID_KOTA)
);
```

Kode Sumber 4.4. Implementasi Tabel Kota

```
create table KATEGORI
(
  ID_KATEGORI      int not null,
  ID_KOTA          int not null,
  NAMA_KATEGORI    varchar(25),
  primary key (ID_KATEGORI)
);
```

Kode Sumber 4.5. Implementasi Tabel Kategori

```
create table WISATA
(
  ID_WISATA        int not null,
  ID_KATEGORI      int not null,
  JUDUL            varchar(30),
  LINK             varchar(50),
  DESKRIPSI        varchar(100),
  FOTO            varchar(50),
  NOMINALBIAYA     int,
  BIAYA            varchar(20),
  LATITUDE         decimal(15),
  LONGTITUDE       decimal(15),
  primary key (ID_WISATA)
);
```

Kode Sumber 4.6. Implementasi Tabel Wisata

4.4.2. Implementasi *Query*

Implementasi *query* adalah implementasi perintah yang digunakan untuk kebutuhan aplikasi pada basis data. Kebutuhan *query* untuk aplikasi perangkat bergerak yaitu *query* pencarian berdasarkan kriteria, *query* pencarian cepat dengan rekomendasi, *query* detail wisata dan *query* pencarian terdekat.

4.4.2.1. Implementasi *Query* Pencarian Berdasarkan Kriteria

Implementasi *query* untuk pencarian berdasarkan kriteria adalah implementasi perintah untuk mendapatkan daftar wisata sesuai dengan kriteria pengguna dari basis data. Diperlukan masukan berupa nilai kota, kategori dan biaya dari pengguna. Implementasi *query* pencarian wisata berdasarkan daerah dapat dilihat pada Kode sumber 4.7

```
SELECT
kot.`IDKOTA`,kat.`IDKATEGORI`,kot.`NAMA KOTA`,kat.`NAMA KATEGORI`,t.`JUDUL`,t.`BIAYA`
FROM `kota` kot
LEFT JOIN `kategori` kat ON kat.`IDKOTA` = kot.`IDKOTA`
LEFT JOIN `tempat wisata` t ON t.`IDKATEGORI` =
kat.`IDKATEGORI`
WHERE kot.`NAMA KOTA` = '$NAMA KOTA' AND
kat.`NAMA KATEGORI` = '$NAMA KATEGORI' AND
t.`BIAYA` = '$BIAYA'"
```

Kode Sumber 4.7. Implementasi *Query* Pencarian Wisata Berdasarkan Daerah

4.4.2.2. Implementasi *Query* Pencarian Cepat dengan Rekomendasi

Implementasi *query* untuk pencarian cepat dengan rekomendasi adalah implementasi perintah untuk mendapatkan data wisata yang diinginkan oleh pengguna dan daftar rekomendasi wisata yang hampir sama kriterianya. Diperlukan masukan dari pengguna berupa kota dan judul wisata. Rekomendasi didapatkan dengan menggunakan metode *Analytic Hierarchy Proses* (AHP).

Implementasi *query* pencarian cepat dengan rekomendasi dapat dilihat pada Kode sumber 4.8.

```
SELECT
kot.`IDKOTA`,kat.`IDKATEGORI`,kot.`NAMAkota`,kat.`NA
MAKATEGORI`,t.`JUDUL`,t.`BIAYA`
FROM `kota` kot
LEFT JOIN `kategori` kat ON kat.`IDKOTA` = kot.`IDKOTA`
LEFT JOIN `tempatwisata` t ON t.`IDKATEGORI` =
kat.`IDKATEGORI`
WHERE kot.`NAMAkota` = '$nama_kota'
AND kat.`IDKATEGORI` IN(SELECT `IDKATEGORI` FROM
`tempatwisata` WHERE `JUDUL` = '$nama_wisata')
AND t.`BIAYA` IN(SELECT `BIAYA` FROM `tempatwisata`
WHERE `JUDUL` = '$nama_wisata')
```

Kode Sumber 4.8. Implementasi *Query* Pencarian Cepat dengan Rekomendasi

4.4.2.3 Implementasi *Query* Detail Wisata

Implementasi *query* untuk detail wisata adalah implementasi perintah untuk mendapatkan data detail wisata dari basis data. Diperlukan masukkan dari pengguna berupa judul wisata. Hasil dari *query* akan didapatkan judul, alamat sumber, deskripsi, gambar dan koordinat wisata. Kode sumber 4.9 merupakan implementasi *query* detail wisata.

```
SELECT * FROM tempatwisata WHERE JUDUL='$JUDUL'
```

Kode Sumber 4.9. Implementasi *Query* Detail Wisata

4.4.2.4 Implementasi *Query* Pencarian Terdekat

Implementasi *query* untuk pencarian terdekat adalah implementasi perintah untuk mencari wisata terdekat dari lokasi pengguna. Perintah melakukan perhitungan dengan rumus Haversine pada koordinat wisata yang tersimpan pada basis data. Diperlukan masukkan dari pengguna berupa kategori, biaya dan koordinat pengguna yang didapatkan secara otomatis. Implementasi *query* pencarian terdekat dapat dilihat pada Kode sumber 4.10.

```

SELECT
kat.`IDKATEGORI`,kat.`NAMAKATEGORI`,t.`JUDUL`,
t.`BIAYA`, t.`KOORDINATX`, t.`KOORDINATY`,
(6371*acos(cos(radians('$KOORDINATX'))*cos(radians(t.KOO
RDINATX))*cos(radians(t.KOORDINATY)-
radians('$KOORDINATY'))+sin(radians('$KOORDINATX'))*si
n(radians(t.KOORDINATX))))AS distance
FROM `kategori` kat
LEFT JOIN `tempatwisata` t ON t.`IDKATEGORI` =
kat.`IDKATEGORI`
WHERE kat.`NAMAKATEGORI`='$NAMAKATEGORI' AND
t.`BIAYA`='$BIAYA' HAVING distance <'5' ORDER BY
distance LIMIT 0,20

```

Kode Sumber 4.10. Implementasi Query Pencarian Terdekat

4.4.3 Impelementasi Webservice

Implementasi *webservice* adalah implemetasi perintah php untuk mendapat nilai yang akan digunakan pada aplikasi perangkat bergerak.

4.4.3.1 Implementasi Webservice Pencarian Berdasarkan Kriteria

Implementasi *webservice* pencarian berdasarkan kriteria adalah implementasi perintah untuk mendapatkan nilai wisata pada basis data yang sesuai dengan kriteria pengguna. Dalam implementasi tersebut, pengguna memilih kota, kategori dan biaya sebagai masukan, kemudian aplikasi perangkat bergerak mengirimkan nilainya pada *webservice* untuk melakukan perintah pencarian wisata dan hasilnya dikirimkan kembali ke perangkat bergerak. Implementasi *webservice* pencarian berdasarkan kriteria dapat dilihat pada kode sumber 4.11.


```

$response = array();
$con = mysql_connect("sql101.phpnet.us", "pn_15531166",
"sodik20101990");
mysql_select_db("pn_15531166_tugasakhir");
$NAMAKOTA = $_GET["nama_kota"];
$NAMAKATEGORI = $_GET["nama_kategori"];
$BIAYA = $_GET["biaya"];
if($NAMAKOTA != "" AND $NAMAKATEGORI != "" AND
$BIAYA != ""){
$result = mysql_query("SELECT
kot.`IDKOTA`,kat.`IDKATEGORI`,kot.`NAMAKOTA`,kat.`NA
MAKATEGORI`,t.`JUDUL`,t.`BIAYA`
FROM `kota` kot
LEFT JOIN `kategori` kat ON kat.`IDKOTA` = kot.`IDKOTA`
LEFT JOIN `tempatwisata` t ON t.`IDKATEGORI` =
kat.`IDKATEGORI`
WHERE kot.`NAMAKOTA` = '$NAMAKOTA' AND
kat.`NAMAKATEGORI`=$NAMAKATEGORI' AND
t.`BIAYA`=$BIAYA'") or die(mysql_error());
if (mysql_num_rows($result) > 0) {
$response["tempatwisata"] = array();
while ($row = mysql_fetch_array($result)) {
$tempatwisata= array();
$tempatwisata["nama_kota"] = $row["NAMAKOTA"];
$tempatwisata["nama_kategori"]= $row["NAMAKATEGORI"];
$tempatwisata["Biaya"] = $row["BIAYA"];
$tempatwisata["Judul"] = $row["JUDUL"];
array_push($response["tempatwisata"], $tempatwisata);
}
$response["success"] = 1;
echo json_encode($response);
}
else {
$response["success"] = 0;
$response["message"] = "data tidak ditemukan";
echo json_encode($response);
}
}
else {

```

```

$response["success"] = 0;
$response["message"] = "parsing data error";
echo json_encode($response);
}

```

Kode Sumber 4.11. Implementasi *Webservice* Pencarian Berdasarkan Kriteria

4.4.3.2 Implementasi *Webservice* Pencarian Cepat dengan Rekomendasi

Implementasi *webservice* pencarian cepat dengan rekomendasi adalah implementasi perintah untuk mendapatkan hasil wisata sesuai dengan wisata yang dimasukkan oleh pengguna. Dalam implementasi tersebut, pengguna memasukkan nama wisata sebagai masukan, kemudian aplikasi perangkat bergerak mengirimkan nilainya pada *webservice* untuk melakukan perintah mendapatkan hasil wisata yang sesuai dengan masukan. *Webservice* selain mendapatkan hasil wisata yang sesuai masukan, juga akan mendapat rekomendasi wisata lain yang hampir sama kriterianya dengan hasil wisata yang telah didapatkan. Implementasi *webservice* pencarian cepat dengan rekomendasi dapat dilihat pada kode sumber 4.12.

```

$response = array();
$con = mysql_connect("sql101.phpnnet.us", "pn_15531166",
"sodik20101990");
mysql_select_db("pn_15531166_tugasakhir");
$nama_kota = $_GET["nama_kota"];
$nama_wisata = $_GET["wisata"];
if($nama_kota != "" AND $nama_wisata != "")
{
$result = mysql_query("SELECT
kot.`IDKOTA`,kat.`IDKATEGORI`,kot.`NAMA KOTA`,kat.`NA
MAKATEGORI`,t.`JUDUL`,t.`BIAYA`

FROM `kota` kot
LEFT JOIN `kategori` kat ON kat.`IDKOTA` = kot.`IDKOTA`
LEFT JOIN `tempatwisata` t ON t.`IDKATEGORI` =

```



```

kat.`IDKATEGORI`
WHERE kot.`NAMA_KOTA` = '$nama_kota' AND
kat.`IDKATEGORI` IN(SELECT `IDKATEGORI`
FROM `tempatwisata` WHERE `JUDUL` = '$nama_wisata')
AND t.`BIAYA` IN(SELECT `BIAYA` FROM
`tempatwisata` WHERE `JUDUL` = '$nama_wisata'))" or
die(mysql_error());
if (mysql_num_rows($result) > 0)
{
$response["tempatwisata"] = array();
while ($row = mysql_fetch_array($result))
{
$tempatwisata = array();
$tempatwisata ["nama_wisata"] = $row["JUDUL"];
$tempatwisata ["nama_kota"] = $row["NAMA_KOTA"];
$tempatwisata ["nama_kategori"] = $row["NAMA_KATEGORI"];
$tempatwisata ["nama_biaya"] = $row["BIAYA"];
array_push($response["tempatwisata"], $tempatwisata);
}
for ($i = 0; $i < count($response["tempatwisata"]); $i++)
{
if($response["tempatwisata"][$i]["nama_wisata"] ==
$nama_wisata)
{
$stemp = $response["tempatwisata"][0];
$response["tempatwisata"][0] = $response["tempatwisata"][$i];
$response["tempatwisata"][$i] = $stemp;
}
}
$response["success"] = 1;
echo json_encode($response);
}
else {
$response["success"] = 0;
$response["message"] = "data tidak ditemukan";
echo json_encode($response);
}
}
else {

```

```

$response["success"] = 0;
$response["message"] = "parsing data error";
echo json_encode($response);
}

```

Kode Sumber 4.12. Implementasi *Webservice* Pencarian Cepat Dengan Rekomendasi

4.4.3.3 Implementasi *Webservice* Detail Wisata

Implementasi *web service* detail wisata adalah implementasi perintah untuk mendapatkan detail dari salah satu wisata. Detail wisata dalam *web service* ini adalah judul, alamat sumber, ulasan dan gambar dari wisata yang telah dipilih. Implementasi *web service* detail wisata dapat dilihat pada kode sumber 4.13.

```

$response = array();
$con = mysql_connect("sql101.phpnet.us", "pn_15531166",
"sodik20101990");
mysql_select_db("pn_15531166_tugasakhir");
$JUDUL = $_GET["judul"];
if($JUDUL != "")
{
$result = mysql_query("SELECT * FROM tempatwisata
WHERE JUDUL='$JUDUL'") or die(mysql_error());
if (mysql_num_rows($result) > 0)
{
$response["tempatwisata"] = array();
while ($row = mysql_fetch_array($result))
{
$tempatwisa= array();
$tempatwisa["Judul"] = $row["JUDUL"];
$tempatwisa["Alamat_sumber"] = $row["LINK"];
$tempatwisa["Deskripsi"] = $row["DESKRIPSI"];
$tempatwisa["Foto"] = $row["FOTO"];
$tempatwisa["Biaya"] = $row["BIAYA"];
array_push($response["tempatwisata"],$tempatwisa);
}
$response["success"] = 1;
echo json_encode($response);
}
}

```



```

else {
    $response["success"] = 0;
    $response["message"] = "data tidak ditemukan";
    echo json_encode($response);
}
}
else {
    $response["success"] = 0;
    $response["message"] = "parsing data error";
    echo json_encode($response);
}
}

```

Kode Sumber 4.13. Implementasi *Webservice* Detail Wisata

4.4.3.4 Implementasi *Webservice* Pencarian Terdekat

Implementasi *webservice* pencarian terdekat adalah implementasi perintah untuk mendapatkan data wisata dengan jarak terdekat dengan lokasi pengguna. Dalam implementasi tersebut diperlukan nilai kategori, biaya dan koordinat pengguna. Implementasi *webservice* pencarian terdekat dapat dilihat pada kode sumber 4.14.

```

<?php
$response = array();
$con = mysql_connect("sql101.phpnet.us", "pn_15531166",
"sodik20101990");
mysql_select_db("pn_15531166_tugasakhir");
$NAMA_kategori = $_GET["nama_kategori"];
$BIAYA = $_GET["biaya"];
$KOORDINATX = $_GET["KoorX"];
$KOORDINATY = $_GET["KoorY"];
if($NAMA_kategori != "" AND $BIAYA != "" AND
$KOORDINATX != "" AND $KOORDINATY != ""){
$result = mysql_query("SELECT
kat.`IDKATEGORI`,kat.`NAMA_kategori`,t.`JUDUL`,
t.`BIAYA`, t.`KOORDINATX`, t.`KOORDINATY`,
(6371*acos(cos(radians('$KOORDINATX'))*cos(radians(t.KOORDINATX))
*cos(radians(t.KOORDINATY)-
radians('$KOORDINATY'))+sin(radians('$KOORDINATX'))*si

```

```

n(radians(t.KOORDINATX))))AS distance
FROM `kategori` kat
LEFT JOIN `tempatwisata` t ON t.`IDKATEGORI` =
kat.`IDKATEGORI`
WHERE kat.`NAMAKATEGORI`=$NAMAKATEGORI AND
t.`BIAYA`=$BIAYA HAVING distance <'5' ORDER BY
distance LIMIT 0,20") or die(mysql_error());
if (mysql_num_rows($result) > 0) {
$response["tempatwisata"] = array();
while ($row = mysql_fetch_array($result)) {
$tempatwisata["Judul"] = $row["JUDUL"];
array_push($response["tempatwisata"], $tempatwisata);
}
$response["success"] = 1;
echo json_encode($response);
}
else {
$response["success"] = 0;
$response["message"] = "data tidak ditemukan";
echo json_encode($response);
}}
else {
$response["success"] = 0;
$response["message"] = "parsing data error";
echo json_encode($response);
}
?>

```

Kode Sumber 4.14. Implementasi Webservice Pencarian Terdekat

4.4.4 Implementasi Perangkat Bergerak (*Mobile*)

Implementasi perangkat bergerak adalah implementasi berbagai fungsi pada aplikasi perangkat bergerak pencarian wisata berdasarkan daerah. Adapun implementasinya adalah implementasi pencarian wisata berdasarkan kriteria, implementasi pencarian cepat dengan rekomendasi, implementasi detail wisata, dan implementasi penunjuk jalan.

4.4.4.1 Implementasi Pencarian Wisata Berdasarkan Kriteria

Implementasi pencarian wisata berdasarkan daerah pada perangkat bergerak adalah implementasi fungsi pencarian wisata sesuai dengan masukan kriteria dari pengguna. Pengguna dapat memilih kota, kategori dan biaya sebagai masukan. Masukan (**valueKota**, **valueKategori**, **valueBiaya**) dari pengguna akan dikirim oleh aplikasi ke *webservice* (**jParser.makeHttpRequest(URL_WisataByKriteria,"GET", params)**), kemudian *webservice* akan menjalankan perintah untuk mendapatkan hasil wisata. Setelah hasil wisata didapatkan, *webservice* mengirimkan hasil kembali ke aplikasi perangkat bergerak untuk ditampilkan ke pengguna. Implementasi pencarian wisata berdasarkan kriteria pada perangkat bergerak java dapat dilihat pada kode sumber 4.15.

```
if(katBiaya == 3){
    valueBiaya = "Murah";
}
else if(katBiaya == 2){
    valueBiaya = "Sedang";
}
else if(katBiaya == 1){
    valueBiaya = "Mahal";
}
// TODO Auto-generated method stub
hasilList = new ArrayList<String>();
List<NameValuePair> params = new
ArrayList<NameValuePair>();
params.add(new BasicNameValuePair(TAG_NAMEKOTA,
valueKota));
params.add(new
BasicNameValuePair(TAG_NAMEKATEGORI,valueKategori));
params.add(new BasicNameValuePair(TAG_NAMEBIAYA,
valueBiaya));
JSONObject json =
jParser.makeHttpRequest(URL_WisataByKriteria, "GET",
params);
```

```

if(json != null) {
    try {
        if(json.getInt("success") == 1){
            JSONArray wisata = json.getJSONArray("tempatwisata");
            for (int i = 0; i < wisata.length(); i++) {
                JSONObject catObj = (JSONObject) wisata.get(i)
                hasilList.add(catObj.getString("Judul"));
            }
        }else{
            hasilList.add("Wisata tidak ditemukan");
        }
    }
    catch (Exception e)
    {
        // TODO: handle exception
        e.printStackTrace();
    }
}

Intent i = new Intent(getApplicationContext(),
    HasilActivity.class);
i.putStringArrayListExtra("list_hasil", hasilList);
i.setFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP |
    Intent.FLAG_ACTIVITY_CLEAR_TOP);
startActivity(i);
return null;

```

**Kode Sumber 4.15. Implementasi Java Pencarian Wisata
Berdasarkan Kriteria**

4.4.4.2 Implementasi Pencarian Cepat Dengan Rekomendasi

Implementasi pencarian cepat dengan rekomendasi pada perangkat bergerak adalah implmentasi fungsi pencarian dengan memasukan wisata yang dicari oleh pengguna. Pengguna memilih kota dan memasukan judul wisata yang ingin di cari sebagai masukan. Nilai masukan kota dan judul wisata dikirim ke *webservice* (**jParser.makeHttpRequest (URL_AHP,"GET",params)**). *Webservice* melakukan perintah pencarian pada basis data dan hasilnya dikirim kembali ke aplikasi

perangkat bergerak. Pada pencarian yang dilakukan oleh *webservice*, *webservice* tidak hanya mengambil hasil wisata yang dicari oleh pengguna, *webservice* juga mengambil wisata yang hampir sama kriterianya yang dicari oleh pengguna. Implementasi pencarian cepat dengan rekomendasi java dapat dilihat pada kode sumber 4.16.

```
@Override
protected Void doInBackground(Void... arg0)
{
    // TODO Auto-generated method stub
    hasilList = new ArrayList<String>();
    List<NameValuePair> params = new
    ArrayList<NameValuePair> ();
    params.add(new BasicNameValuePair(TAG_NAMEKOTA,
    ValueKota));
    params.add(new BasicNameValuePair(TAG_NAMEWISATA,
    wisataText.getText().toString().toLowerCase()));
    JSONObject json =
    jParser.makeHttpRequest(URL_AHP,"GET",params );
    try {
        flag1 = json.getBoolean("flag");
        if (json.getInt("success") == 1){
            flagList = true;
            Log.e("Response: ", "> " + json.toString());
        }
        try {
            JSONArray wisata = json.getJSONArray("tempatwisata");
            for (int i = 0; i < wisata.length(); i++) { JSONObject catObj =
            (JSONObject) wisata.get(i);
            hasilList.add(catObj.getString("nama_wisata"));
            }
        }
        catch (JSONException e) {
            e.printStackTrace();
        }
        else {
            flagList = false;
            Log.e("JSON Data", "Didn't receive any data from server!");
        }
    }
}
```

```

flag = true;
} catch (JSONException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
return null;
}

```

Kode Sumber 4.16. Implementasi Java Pencarian Cepat dengan Rekomendasi

4.4.4.3 Implementasi Detail Wisata

Implementasi detail wisata pada perangkat bergerak adalah implementasi fungsi untuk menampilkan detail wisata yang telah dipilih oleh pengguna. Pengguna setelah melakukan pencarian kemudian memilih satu wisata untuk melihat detail wisata tersebut. Pilihan wisata yang dilakukan oleh pengguna adalah sebuah masukan ("judul",idWisata) untuk dikirim ke *webservice*(*jsonParser.makeHttpRequest(URL_detailWisata,"GET" , params)*). *Webservice* mengambil detail wisata pada basis data untuk dikirimkan ke aplikasi perangkat bergerak. Adapun hasil yang dikirimkan ke perangkat bergerak adalah judul, sumber data, ulasan dan gambar. Implementasi detail wisata java dapat dilihat pada kode sumber 4.17.

```

String idWisata = jdlW;
List<NameValuePair> params = new ArrayList
<NameValuePair>();
params.add(new BasicNameValuePair("judul",idWisata));
try
{
JSONObject json =
jsonParser.makeHttpRequest(URL_detailWisata,"GET" , params);
Log.d("Single Product Details", json.toString());
int success = json.getInt(TAG_SUCCESS);
if (success == 1) {
// successfully received product details
JSONArray productObj = json.getJSONArray("tempatwisata");
WisataKu = productObj.getJSONObject(0);
}
}

```



```

    }
    else{
        // product with pid not found
    }
}
catch (JSONException e) {
    e.printStackTrace();
}
return null;

```

Kode Sumber 4.17. Implementasi Java Detail Wisata

4.4.4.4 Implementasi Penunjuk Jalan Dengan Google Maps

Implementasi penunjuk jalan dengan Google Maps adalah implementasi fungsi penunjuk jalan dengan menggunakan aplikasi Google Maps untuk membantu pengguna aplikasi WisataINA menemukan jalan dari lokasi sekarang ini ke wisata yang dituju. Untuk menggunakan fungsi penunjuk jalan, pengguna setelah melihat detail wisata, kemudian menekan tombol penunjuk jalan. Data yang tersimpan pada detail wisata sudah terdapat koordinat lokasi wisata (**ValueKoorX**, **ValueKoorY**). Implementasi penunjuk jalan dengan Google Maps dapat dilihat pada kode sumber 4.18.

```

Intent intent = new Intent( Intent.ACTION_VIEW,
Uri.parse("http://ditu.google.cn/maps?f=d&source=s_d"+"&da
ddr="+ValueKoorX+" "+ValueKoorY+"&hl=zh&t=m&dirflg=
d"));
intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK&Inten
t.FLAG_ACTIVITY_EXCLUDE_FROM_RECENTS);
intent.setClassName("com.google.android.apps.maps",
"com.google.android.maps.MapActivity");
startActivity(intent);

```

Kode Sumber 4.18. Implementasi Java Penunjuk Jalan menggunakan Google Maps

4.4.4.5 Implementasi Pencarian Terdekat

Implementasi pencarian terdekat adalah implementasi fungsi pencarian wisata dengan lokasi terdekat dari pengguna. Jarak lokasi pencarian pencarian terdekat ini maksimal 10 km diukur dari posisi pengguna.

Dalam melakukan pencarian wisata terdekat, pengguna perlu memilih kategori dan kriteria biaya, untuk lokasi aplikasi langsung mendeteksi posisi pengguna sekarang ini. kategori (**valueKategori**), kriteria biaya (**valueBiaya**) dan koordinat pengguna adalah masukan yang dikirim ke *webservice* (**jParser.makeHttpRequest(URL_Terdekat, "GET", params)**) untuk mendapatkan hasil wisata. Implementasi pencarian terdekat pada perangkat bergerak java dapat dilihat pada kode sumber 4.19.

```
private class buttonCari extends AsyncTask<Void, Void, Void>
{
    @Override
    protected Void doInBackground(Void... arg0) {
        if(indexBiaya == 3){
            valueBiaya = "Murah";
        }
        else if(indexBiaya == 2){
            valueBiaya = "Sedang";
        }
        else if(indexBiaya == 1){
            valueBiaya = "Mahal";
        }
        hasilList = new ArrayList<String>();
        List<NameValuePair> params = new
        ArrayList<NameValuePair>();
        params.add(new
        BasicNameValuePair(TAG_NAMEKATEGORI,
        valueKategori));
        params.add(new BasicNameValuePair(TAG_NAMEBIAYA,
        valueBiaya));
        params.add(new BasicNameValuePair(TAG_KOORDINATX,
        KoorX.toString()));
        params.add(new BasicNameValuePair(TAG_KOORDINATY,
```



```

KoorY.toString());
JSONObject json = jParser.makeHttpRequest(URL_Terdekat,
"GET", params);
if(json != null) {
Log.d("Semua wisata:", json.toString());
try {
if(json.getInt("success") == 1){
JSONArray wisata = json.getJSONArray("tempatwisata");
for (int i = 0; i < wisata.length(); i++) {
JSONObject catObj = (JSONObject) wisata.get(i);
hasilList.add(catObj.getString("Judul"));
}
}
else{
hasilList.add("Wisata tidak ditemukan");
}
}
catch (Exception e) {
// TODO: handle exception
e.printStackTrace();
}
}
Intent i = new Intent(getApplicationContext(),
HasilActivity.class);
i.putStringArrayListExtra("list_hasil", hasilList);
i.setFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP |
Intent.FLAG_ACTIVITY_CLEAR_TOP);
startActivity(i);
return null;
}

```

Kode Sumber 4.19. Implementasi Java Pencarian Terdekat

4.4.5 Implementasi Aplikasi *Website*

Implementasi aplikasi *website* adalah implementasi berbagai fungsi pada aplikasi *website* yang berfungsi untuk mengelola data aplikasi pencarian wisata berdasarkan daerah yang dilakukan oleh administrator. Adapun fitur aplikasi *website* adalah

pencarian, tambah wisata manual, tambah wisata dengan *scraping*, pengaturan provinsi, pengaturan kota, pengaturan kategori dan pengaturan wisata.

4.4.5.1 Implementasi Melihat Wisata

Implementasi melihat wisata adalah implementasi fungsi untuk mencari dan melihat wisata yang ada pada basis data. Untuk melihat wisata, pengguna harus memilih provinsi, kota dan kategori wisata dan kemudian menekan tombol pilih. Pilihan pengguna adalah masukan dari pengguna yang digunakan untuk filter data wisata yang ditampilkan. Aplikasi *website* akan menampilkan hasil wisata sesuai dengan kriteria yang telah dipilih oleh pengguna. Atribut wisata yang ditampilkan adalah judul ('**name**'=>'JUDUL'), ulasan(**name**'=>'DESKRIPSI'), nominalbiaya(**name**'=>'NOMINALBIAYA'), biaya ('**name**'=>'BIAYA') dan gambar(**name**'=>'FOTO). Implementasi pencarian wisata *website controller php* dapat dilihat pada kode sumber 4.20.

```
public function actionAjaxfilter(){
    $criteria1 = new CDbCriteria;
    $criteria2 = new CDbCriteria;
    $array = "";
    if(isset($_POST['IDPROVINST']) &&
    $_POST['IDKATEGORI']){
        if($_POST['IDKATEGORI']!=0){
            if(strlen($criteria1->condition))
            $criteria1->condition .= ' AND';
            $criteria1->condition .=
            IDKATEGORI='$_POST['IDKATEGORI'];
        }
    }
    else
    $criteria1->condition = 'IDTEMPATWISATA = 0';
    $dataProvider=new CActiveDataProvider('Tempatwisata',array(
    'criteria' => $criteria1,
    'pagination'=>array('pageSize'=>50)
    ));
    $this->widget('zii.widgets.grid.CGridView', array(
    'id'=>'file-grid',
```



```
'dataProvider'=>$dataProvider,
'enableSorting'=> false,
'columns'=>array(
array(
'name'=>'JUDUL',
'htmlOptions'=>array('width'=>'10%'),
),
array(
'name'=>'DESKRIPSI',
'htmlOptions'=>array('width'=>'40%'),
),
array(
'name'=>'NOMINALBIAYA',
'htmlOptions'=>array('width'=>'10%'),
),
array(
'name'=>'BIAYA',
'htmlOptions'=>array('width'=>'5%'),
),
array(
'name'=>'FOTO',
'type'=>'image',
'value'=>strpos($data->FOTO,"http:")?$data->FOTO:"images/".$data->FOTO',
'htmlOptions'=>array('width'=>'30%', 'height'=>'200px'),
),),); }
```

Kode Sumber 4.20. Implementasi *Controller* Melihat Wisata

4.4.5.2 Implementasi Tambah Wisata Manual

Implementasi tambah wisata manual adalah implementasi fungsi tambah data wisata dengan administrator menuliskan judul (`$_POST['judul']`), alamat sumber, ulasan (`$_POST['deskripsi']`), dan gambar (`$_FILES['gambar']`) secara manual. Administrator menuliskan informasi kedalam halaman tambah wisata manual. Implementasi *controller* php tambah wisata manual dapat dilihat pada kode sumber 4.21.

```

public function actiontambahWisata(){
    $criteria = new CDbCriteria;
    $model = CHtml::listData(Provinsi::model()-
    >findAll($criteria),'IDPROVINSI','NAMAPROVINSI');
    if(isset($_FILES['gambar']) &&
    count($_FILES['gambar']['error']) == 1 &&
    $_FILES['gambar']['error'][0] > 0){
        echo("<script>window.alert('Foto harus diisi');</script>");
        $this->render('tambahWisata', array('model'=>$model));
    }
    else if(isset($_FILES['gambar'])){
        if($_POST['IDPROVINSI']==0){
            echo "<script>alert('Mohon pilih provinsi terlebih dahulu');
            </script>";
            $this->redirect(array('tambahWisata'));
        }
        if($_POST['IDKOTA']==null||$_POST['IDKOTA']==0){
            echo "<script>alert('Mohon pilih kota terlebih dahulu');
            </script>";
            $this->redirect(array('tambahWisata'));
        }
        if($_POST['IDKATEGORI']==null||$_POST['IDKATEGORI']==
        0){
            echo "<script>alert('Mohon pilih kategori terlebih
            dahulu');</script>";
            $this->redirect(array('tambahWisata'));
        }
        $tempatwisata = new Tempatwisata();
        $tempatwisata->IDKATEGORI = $_POST["IDKATEGORI"];
        $tempatwisata->JUDUL = strtolower($_POST["judul"]);
        $tempatwisata->DESKRIPSI = $_POST["deskripsi"];
        $tempatwisata->LINK = $_POST['alamat_web'];
        $tempatwisata->FOTO = $_FILES['gambar']['name'][0];
        move_uploaded_file($_FILES['gambar']['tmp_name'][0],
        Yii::app()->params['filedir'].$_FILES['gambar']['name'][0]);
        echo Yii::app()->params['filedir'].$_FILES['gambar']['name'][0];
        $tempatwisata->NOMINALBIAYA = $_POST["nominalbiaya"];
        if($_POST["nominalbiaya"] <= 50000){
            $tempatwisata->BIAYA = "Murah";
        }
    }
}

```



```

if($_POST["hotel"] > 50000 && $_POST["hotel"] <= 150000){
    $tempatwisata->BIAYA = "Sedang";
}
if($_POST["hotel"] > 150000){
    $tempatwisata->BIAYA = "Mahal";
}
$tempatwisata->KOORDINATX = (float)$_POST["koorX"];
$tempatwisata->KOORDINATY = (float)$_POST["koorY"];
if($tempatwisata->save()){
    $this->redirect(array('/tempatwisata/ view','id'=>$tempatwisata-
    >IDTEMPATWISATA));
} }
else
    $this->render('tambahWisata', array('model'=>$model));
}

```

Kode Sumber 4.21. Implementasi Controller Tambah Wisata Manual

4.4.5.3 Implementasi Tambah Wisata *Website Scraping*

Implementasi tambah wisata *website scraping* adalah implementasi fungsi untuk menambah data wisata dengan *scraping website* wisata yang lain. Penambahan data dengan *scraping* (**public function scrape_between(\$data, \$start, \$end)**) dilakukan dengan cara mengunduh data yang dibutuhkan dan kemudian disimpan didalam basis data. Data yang di-*scrap* adalah judul, ulasan dan gambar. Implementasi fungsi *scraping* dapat dilihat pada kode sumber 4.22 dan Implementasi *controller* tambah wisata *website scraping* dapat dilihat pada kode sumber 4.23.

```

function curl_exec_follow($ch, &$maxredirect = null) {
    $user_agent = "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-
    US; rv:1.7.5)." Gecko/20041107 Firefox/1.0";
    curl_setopt($ch, CURLOPT_USERAGENT, $user_agent );
    $mr = $maxredirect === null ? 5 : intval($maxredirect);
    if (ini_get('open_basedir') == " && ini_get('safe_mode') == 'off')
    {
        curl_setopt($ch, CURLOPT_FOLLOWLOCATION, $mr > 0);
        curl_setopt($ch, CURLOPT_MAXREDIRS, $mr);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    }
}

```

```

curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
} else {
curl_setopt($ch, CURLOPT_FOLLOWLOCATION, false);
if ($mr > 0){
$original_url = curl_getinfo($ch,
CURLOPT_EFFECTIVE_URL);
$newurl = $original_url;
$rch = curl_copy_handle($ch);
curl_setopt($rch, CURLOPT_HEADER, true);
curl_setopt($rch, CURLOPT_NOBODY, true);
curl_setopt($rch, CURLOPT_FORBID_REUSE, false);
do
{
curl_setopt($rch, CURLOPT_URL, $newurl);
$header = curl_exec($rch);
if (curl_errno($rch))
{
$code = 0;
} else{
$code = curl_getinfo($rch, CURLOPT_HTTP_CODE);
if ($code == 301 || $code == 302) {
preg_match('/Location:(.*?)\n/i', $header, $matches);
$newurl = trim(array_pop($matches));
if(!preg_match("/^https?:/i", $newurl)){
$newurl = $original_url . $newurl;
} }
} else
{
$code = 0;
}}}
while ($code && --$mr);
curl_close($rch);
if (!$mr)
{
if ($maxredirect === null)
trigger_error("Too many redirects.", E_USER_WARNING);
else
$maxredirect = 0;
return false;
}

```



```

}
curl_setopt($ch, CURLOPT_URL, $newurl);
}}
return curl_exec($ch);
}

$ch = curl_init($url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
ini_set('max_execution_time', 0);
$data = $this->curl_exec_follow($ch);
curl_close($ch);
return $data;
}

public function scrape_between($data, $start, $end){
    $data = strstr($data, $start);
    $data = substr($data, strlen($start));
    $stop = strpos($data, $end);
    $data = substr($data, 0, $stop);
    return $data;
}

```

Kode Sumber 4.22. Implementasi fungsi *Scraping*

```

public function actiontambahWeb(){
    $criteria = new CDbCriteria;
    $model = CHtml::listData(Provinsi::model()-
    >findAll($criteria), 'IDPROVINSI', 'NAMAPROVINSI');
    if(isset($_POST['IDPROVINSI'])){
        if($_POST['IDPROVINSI']==0){
            echo '<script>alert("Mohon pilih provinsi terlebih
            dahulu");</script>';
            $this->redirect(array('tambahWeb'));
        }
        if($_POST['IDKOTA']==null||$_POST['IDKOTA']==0){
            echo '<script>alert("Mohon pilih kota terlebih dahulu");</script>';
            $this->redirect(array('tambahWeb'));
        }
        if($_POST['IDKATEGORI']==null||$_POST['IDKATEGORI']==

```

```

0)
{
echo '<script>alert("Mohon pilih kategori terlebih
dahulu");</script>';
$this->redirect(array('tambahWeb'));
}
$judul_start = $_POST["judul_start"];
$judul_end = $_POST["judul_end"];
$deskripsi_start = $_POST["deskripsi_start"];
$deskripsi_end = $_POST["deskripsi_end"];
$foto_link = $_POST["foto_link"];
$foto_start = $_POST["foto_start"];
$foto_end = $_POST["foto_end"];
$sumber = $this->curl($_POST['alamat_web']);
$judul = $this->scrape_between($sumber, $judul_start,
$judul_end);
$deskripsi = $this->scrape_between($sumber, $deskripsi_start,
$deskripsi_end);
$image = $this->scrape_between($sumber, $foto_start,
$foto_end);
$imageurl = $foto_link.$image;
$tempatwisata = new Tempatwisata();
$tempatwisata->IDKATEGORI = $_POST['IDKATEGORI'];
$tempatwisata->LINK = $_POST['alamat_web'];
$tempatwisata->JUDUL = strtolower($judul);
$tempatwisata->DESKRIPSI = $deskripsi;
$tempatwisata->FOTO = $imageurl;
$tempatwisata->NOMINALBIAYA = $_POST["hotel"];
if($_POST["hotel"] <= 50000){
$tempatwisata->BIAYA = "Murah";
}
if($_POST["hotel"] > 50000 && $_POST["hotel"] <= 150000){
$tempatwisata->BIAYA = "Sedang";
}
if($_POST["hotel"] > 150000){
$tempatwisata->BIAYA = "Mahal";
}
$tempatwisata->KOORDINATX = (float)$_POST["koorX"];
$tempatwisata->KOORDINATY = (float)$_POST["koorY"];

```



```

if($tempatwisata->save()){
    $this->redirect(array('/tempatwisata/view','id'=> $tempatwisata-
    >IDTEMPATWISATA));
} }
else
    $this->render('tambahWeb', array('model'=>$model));
}

```

Kode Sumber 4.23. Implementasi *Controller* Tambah Wisata Scraping Website

4.4.5.4 Implementasi Pengaturan Kota

Implementasi pengaturan kota adalah implementasi fungsi untuk mengelola data kota. Mengelola data kota meliputi tambah (**public function actionCreate()**), hapus (**public function actionDelete(\$id)**) dan rubah (**public function actionUpdate(\$id)**). Implementasi *controller* php pengaturan kota dapat dilihat pada kode sumber 4.24.

```

<?php
class KotaController extends Controller{
    public $layout='../layouts/column2';
    public function filters(){
        return array(
            'accessControl',
            'postOnly + delete',
        );
    }
    public function accessRules(){
        return array (
            array('allow',
                'users'=>array('@'),
            ),
            array('deny',
                'users'=>array('*'),
            ), );
    }
    public function actionView($id){
        $this->render('view',array(
            'model'=>$this->loadModel($id),
        ));
    }
    public function actionCreate(){

```

```

$model=new Kota;
if(isset($_POST['Kota'])){
    $model->attributes=$_POST['Kota'];
    if($model->save())
        $this->redirect(array('view','id'=>$model->IDKOTA));
    }
    $this->render('create',array(
        'model'=>$model,
    ));
    public function actionUpdate($id){
        $model=$this->loadModel($id);
        if(isset($_POST['Kota'])){
            $model->attributes=$_POST['Kota'];
            if($model->save())
                $this->redirect(array('view','id'=>$model->IDKOTA));
            }
            $this->render('update',array(
                'model'=>$model,
            )); }
            public function actionDelete($id){
                $this->loadModel($id)->delete();
                if(!isset($_GET['ajax']))
                    $this->redirect(isset($_POST['returnUrl']) ? $_POST['returnUrl']
                        : array('admin'));
                }
                public function actionIndex(){
                    $dataProvider=new CActiveDataProvider('Kota');
                    $this->render('index',array(
                        'dataProvider'=>$dataProvider,
                    ));
                    }
                    public function actionAdmin(){
                        $model=new Kota('search');
                        $model->unsetAttributes();
                        if(isset($_GET['Kota']))
                            $model->attributes=$_GET['Kota'];
                        $this->render('admin',array(
                            'model'=>$model,
                        ));
                    }
                
```



```

}
public function loadModel($id){
$model=Kota::model()->findByPk($id);
if($model===null)
throw new CHttpException(404,'The requested page does not
exist.');
```

```

return $model;
}
protected function performAjaxValidation($model){
if(isset($_POST['ajax']) && $_POST['ajax']=='kota-form'){
echo CActiveForm::validate($model);
Yii::app()->end();
} }
public function actionDynamicdd(){
if(isset($_POST['IDPROVINSI'])){
$data = Kategori::model()->findAll('IDPROVINSI =
'.(int)$_POST['IDPROVINSI']);
$data =
CHtml::listData($data,'IDKATEGORI','NAMAKATEGORI');
}
else
return;
foreach($data as $value=>$name){
echo CHtml::tag('option',
array('value'=>$value),CHtml::encode($name),true);
} } }
```

Kode Sumber 4.24. Implementasi *Controller* Pengaturan Kota

4.4.5.5 Implementasi Pengaturan Kategori

Implementasi pengaturan kategori adalah implementasi fungsi untuk mengelola data kategori. Mengelola data kategori meliputi tambah (**public function actionCreate()**), hapus (**public function actionDelete(\$id)**) dan rubah (**public function actionUpdate(\$id)**). Implementasi *controller* php pengaturan kategori dapat dilihat pada kode sumber 4.25.

```

<?php
class KategoriController extends Controller{
public $layout='../layouts/column2';
```

```

public function filters(){
return array(
'accessControl',
'postOnly + delete',
); }
public function accessRules() {
return array(
array('allow',
'users'=>array('*'),
), ); }
public function actionView($id){
$this->render('view',array(
'model'=>$this->loadModel($id),
)); }
public function actionCreate(){
$model=new Kategori;
if(isset($_POST['Kategori'])){
$model->attributes=$_POST['Kategori'];
if($model->save())
$this->redirect(array('view','id'=>$model->IDKATEGORI));
}
$this->render('create',array(
'model'=>$model,
)); }
public function actionUpdate($id){
$model=$this->loadModel($id);
if(isset($_POST['Kategori'])){
$model->attributes=$_POST['Kategori'];
if($model->save())
$this->redirect(array('view','id'=>$model->IDKATEGORI));}
$this->render('update',array(
'model'=>$model,));}
public function actionDelete($id){
$this->loadModel($id)->delete();
if(!isset($_GET['ajax']))
$this->redirect(isset($_POST['returnUrl']) ? $_POST['returnUrl']
: array('admin'));
}
public function actionIndex(){

```



```

$dataProvider=new CActiveDataProvider('Kategori');
$this->render('index',array(
    'dataProvider'=>$dataProvider,
));
}
public function actionAdmin(){
    $model=new Kategori('search');
    $model->unsetAttributes();
    if(isset($_GET['Kategori']))
        $model->attributes=$_GET['Kategori'];
    $this->render('admin',array(
        'model'=>$model,
    ));
}
public function loadModel($id){
    $model=Kategori::model()->findByPk($id);
    if($model===null)
        throw new CHttpException(404,'The requested page does not exist.');
```

return \$model;

```

}
protected function performAjaxValidation($model){
    if(isset($_POST['ajax']) && $_POST['ajax']=== 'kategori-form'){
        echo CActiveForm::validate($model);
        Yii::app()->end();
    }
}
public function actionDynamicdd2(){
    if(isset($_POST['IDPROVINSI'])){
        $data = Kota::model()->findAll('IDPROVINSI =
        '.(int)$_POST['IDPROVINSI']);
        $data = CHtml::listData($data,'IDKOTA','NAMA KOTA');
        $prompt = '-Semua-';
    }
    else if(isset($_POST['IDKOTA'])){
        $data = Kategori::model()->findAll('IDKOTA =
        '.(int)$_POST['IDKOTA']);
        $data =
        CHtml::listData($data,'IDKATEGORI','NAMA KATEGORI');
        $prompt = '-Semua-';
    }
}

```

```

else
return;
echo CHtml::tag('option', array('value'=>0), $prompt,true);
foreach($data as $value=>$name) {
echo CHtml::tag('option',
array('value'=>$value),CHtml::encode($name),true);
} } }

```

Kode Sumber 4.25. Implementasi *Controller* Pengaturan Kategori

4.4.5.6 Implementasi Pengaturan Wisata

Implementasi pengaturan wisata adalah implementasi fungsi untuk mengelola data wisata. Mengelola data wisata meliputi tambah(**public function actionCreate()**), hapus (**public function actionDelete(\$id)**) dan rubah (**public function actionUpdate(\$id)**). Implementasi *controller* php pengaturan wisata dapat dilihat pada kode sumber 4.26.

```

<?php
class TempatkanwisataController extends Controller{
public $layout='//layouts/column2';
public function filters(){
return array(
'accessControl',
'postOnly + delete',
); }
public function accessRules(){
return array(
array('allow',
'users'=>array('@'),
),
array('deny',
'users'=>array('*'),
), ); }
public function actionView($id) {
$this->render('view',array(
'model'=>$this->loadModel($id),
)); }
public function actionCreate(){

```



```

$model=new Tempatwisata;
if(isset($_POST['Tempatwisata']))
{
$model->attributes=$_POST['Tempatwisata'];
if($model->save())
$this->redirect(array('view','id'=>$model-
>IDTEMPATWISATA));
}
$this->render('create',array(
'model'=>$model,
)); }
public function actionUpdate($id) {
$model=$this->loadModel($id);
if(isset($_POST['Tempatwisata'])){
$model->attributes=$_POST['Tempatwisata'];
if($model->NOMINALBIAYA < 50000){
$model->BIAYA = "Murah";
}
if($model->NOMINALBIAYA > 50001 && $model-
>NOMINALBIAYA < 150000){
$model->BIAYA = "Sedang";
}
if($model->NOMINALBIAYA > 150001){
$model->BIAYA = "Mahal";
}
if( $model->FOTO != NULL){
if(Yii::app()->params['filedir'].$model->FOTO === NULL){
if($model->save())
$this->redirect(array('view','id'=>$model-
>IDTEMPATWISATA));
}
else{
if(!strpos($model->FOTO,"ttp:")){
if(isset($_FILES['gambar']) &&
count($_FILES['gambar']['error']) == 1 &&
$_FILES['gambar']['error'][0] > 0){
echo("<script>window.alert('Foto harus diisi');</script>");
$this->render('update',array(
'model'=>$model,

```

```

)); }
else if (isset($_FILES['gambar']))
{
    unlink(Yii::app()->params['filedir'].$model->FOTO);
    $model->FOTO = $_FILES['gambar']['name'][0];
    move_uploaded_file($_FILES['gambar']['tmp_name'][0],
    Yii::app()->params['filedir'].$_FILES['gambar']['name'][0]);
} }
if($model->save())
$this->redirect(array('view','id'=>$model-
>IDTEMPATWISATA));
} }
else{
$model->FOTO = $_FILES['gambar']['name'][0];
move_uploaded_file($_FILES['gambar']['tmp_name'][0],
Yii::app()->params['filedir'].$_FILES['gambar']['name'][0]);
if($model->save())
$this->redirect(array('view','id'=>$model-
>IDTEMPATWISATA));
} }
$this->render('update',array(
'model'=>$model,
)); }
public function actionDelete($id){
$model = $this->loadModel($id);
if( $model->FOTO != NULL){
if(Yii::app()->params['filedir'].$model->FOTO === NULL){
$model->delete();
}
else {
if(!strpos($model->FOTO,"ttp:"))
unlink(Yii::app()->params['filedir'].$model->FOTO);
$model->delete();
} }
else {
$model->delete();
}
if(!isset($_GET['ajax']))
$this->redirect(isset($_POST['returnUrl']) ? $_POST['returnUrl']

```



```

: array('admin'));
}
public function actionIndex(){
    $dataProvider=new CActiveDataProvider('Tempatwisata');
    $this->render('index',array(
        'dataProvider'=>$dataProvider,
    )); }
public function actionAdmin(){
    $model=new Tempatwisata('search');
    $model->unsetAttributes();
    if(isset($_GET['Tempatwisata']))
    $model->attributes=$_GET['Tempatwisata'];
    $this->render('admin',array(
        'model'=>$model,
    )); }
public function loadModel($id){
    $model=Tempatwisata::model()->findByPk($id);
    if($model===null)
    throw new CHttpException(404,'The requested page does not
    exist.');
```

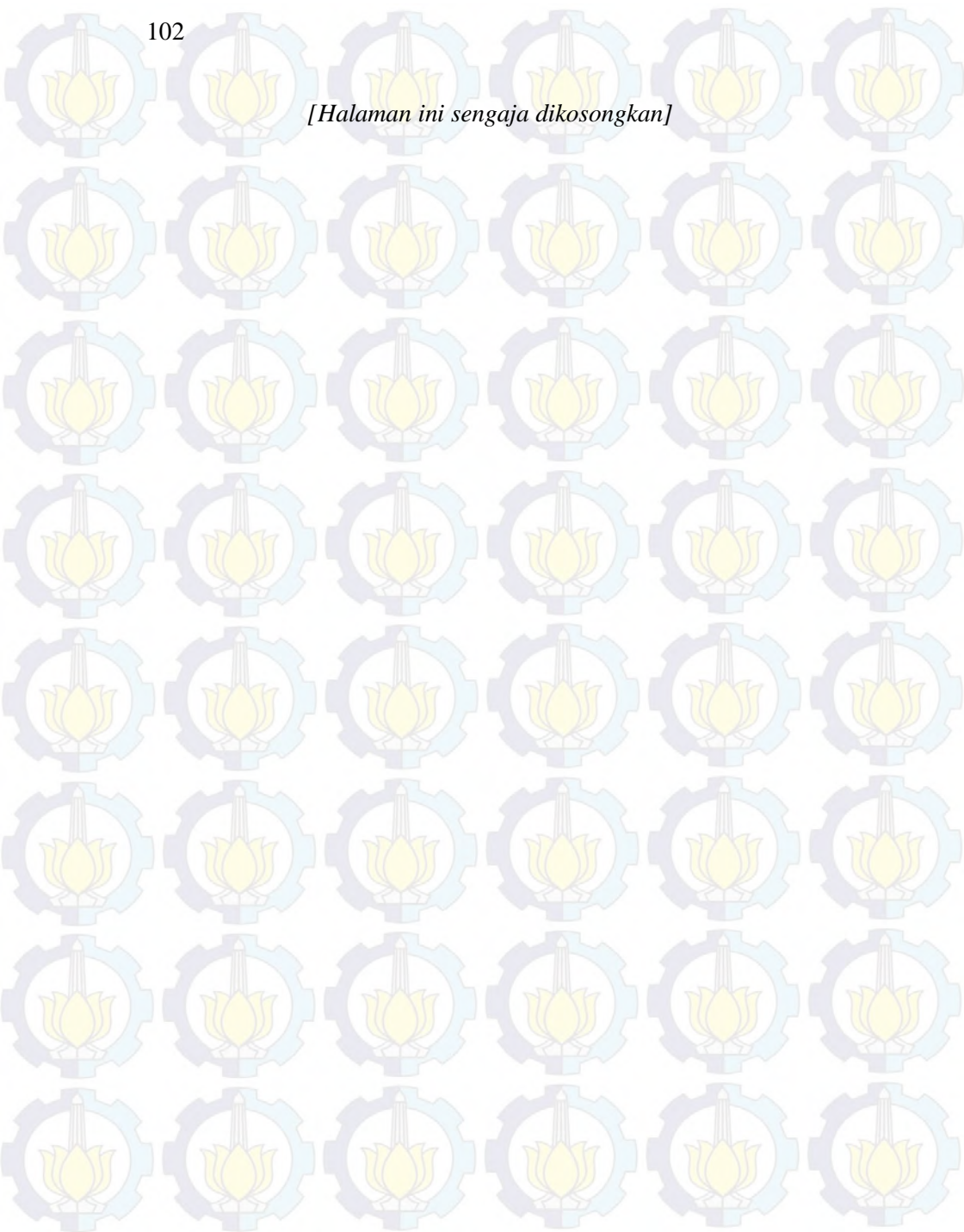
```

    return $model;
}
protected function performAjaxValidation($model){
    if(isset($_POST['ajax']) && $_POST['ajax']==='tempatwisata-
    form'){
        echo CActiveForm::validate($model);
        Yii::app()->end();
    } } }

```

Kode Sumber 4.26. Implementasi *Controller* Pengaturan Wisa

[Halaman ini sengaja dikosongkan]



BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dan evaluasi pada sistem informasi yang dikembangkan. Pengujian yang dilakukan adalah pengujian terhadap kebutuhan fungsionalitas sistem dan kegunaan sistem. Pengujian fungsionalitas mengacu pada kasus penggunaan pada bab tiga. Pengujian kegunaan program dilakukan dengan mengetahui tanggapan dari pengguna terhadap sistem.

5.1 Lingkungan Pengujian

Lingkungan pengujian sistem pada pengerjaan Tugas Akhir ini dilakukan pada lingkungan dan alat kakas seperti pada Tabel 5.1

Tabel 5.1. Lingkungan Pengujian Perangkat Lunak

Laptop	Tipe	Toshiba Satellite L645
	Prosesor	Intel Core i3-M370 CPU @ 2.4GHz(4 CPUs)
	Memori	4096MB RAM
	Sistem Operasi	Windows 7 Home Premium 32-bit
Telepon Pinter	Tipe	Asus Zenfone 5
	Prosesor	Dual-core 1.8 GHz Cortex-A7
	Memori	2048MB RAM
	Sistem Operasi	Android OS, v4.4.2 (Kitkat)
Kakas Bantu	Eclipse Platform	Version: 4.4.2
	Android Developer Tools	Build: v22.3.0
	XAMPP	v3.2.1

5.2. Dasar Pengujian

Pengujian pada rancang bangun aplikasi pencarian wisata berdasarkan daerah dilakukan dengan menggunakan sebuah laptop dan ponsel pintar. Pengujian perangkat lunak ini menggunakan metode pengujian kotak hitam yang berfokus pada kebutuhan fungsional. Pengujian ini dilakukan untuk menguji apakah

fungsionalitas yang diidentifikasi pada tahap kebutuhan benar-benar diimplementasi dan bekerja seperti yang semestinya.

5.3. Pengujian Fungsional

Pada subbab ini dijelaskan mengenai skenario pengujian fungsional dari rancang bangun aplikasi rekomendasi pencarian wisata. Pengujian difokuskan pada uji fungsional. Pengujian yang dilakukan direpresentasikan pada tabel 5.2.

Tabel 5.2. Pengujian Fungsional

Pengguna	Uji Coba	Status
Pengguna Aplikasi	Melakukan Pencarian Berdasarkan Kriteria	Berhasil
	Melakukan Pencarian Cepat dengan Rekomendasi	Berhasil
	Melakukan Pencarian Wisata Terdekat	Berhasil
	Menampilkan Detail Wisata	Berhasil
	Menampilkan Penunjuk Jalan	Berhasil
Pengguna Admin	Melihat Wisata Sesuai Kriteria	Berhasil
	Menambah Wisata Manual	Berhasil
	Menambah Wisata dengan <i>Scrapping Website</i> lain	Berhasil
	Mengelola Provinsi	Berhasil
	Mengelola Kota	Berhasil
	Mengelola Kategori	Berhasil
	Mengelola Wisata	Berhasil

5.3.1. Pengujian Pengguna Aplikasi

Pada pengujian pengguna aplikasi, diuji tentang kehandalan fungsi atau yang berkenaan dengan kebutuhan pengguna aplikasi.

5.3.1.1. Pengujian Pencarian Berdasarkan Kriteria

Skenario pengujian ini merupakan skenario pengujian untuk mengetahui fungsionalitas aplikasi pada fitur pencarian

berdasarkan kriteria. Skenario pengujian dijelaskan pada Tabel 5.3. Tampilan pengujian diperlihatkan pada Gambar 5.1.

Tabel 5.3. Pengujian pencarian berdasarkan kriteria

Nomor	PW01
Nama	Pencarian Wisata Berdasarkan Kriteria
Use Case	F-0001-1
Tujuan	Mendapatkan daftar tempat wisata yang sesuai dengan kriteria pengguna.
Kondisi awal	Kriteria wisata belum dipilih
Skenario	<ol style="list-style-type: none"> 1. Pengguna memilih menu pencarian wisata sesuai dengan kriteria. 2. Pengguna memilih kota, kategori dan biaya masuk tempat wisata 3. Pengguna menekan tombol cari
Masukan	Kriteria kota, kategori dan biaya masuk
Keluaran yang diharapkan	Daftar tempat wisata yang sesuai dengan kriteria pengguna.
Hasil Pengujian	Berhasil



Gambar 5.1 Halaman Pencarian Berdasarkan Kriteria

5.3.1.2. Pengujian Pencarian Cepat

Skenario pengujian ini merupakan skenario pengujian untuk mengetahui fungsionalitas aplikasi pada fitur pencarian cepat dengan rekomendasi. Skenario pengujian dijelaskan pada Tabel 5.4. Tampilan pengujian diperlihatkan pada Gambar 5.2.

Tabel 5.4. Pengujian Pencarian Cepat

Nomor	PW02
Nama	Pencarian cepat dengan rekomendasi
Use Case	F-0001-2
Tujuan	Mendapatkan wisata yang dicari oleh pengguna dan memberikan rekomendasi wisata yang lain
Kondisi awal	Kota belum dipilih dan judul wisata belum di masukkan
Skenario	<ol style="list-style-type: none">1. Pengguna memilih menu pencarian cepat.2. Pengguna memilih kota dan memasukkan judul wisata3. Pengguna menekan tombol cari
Masukan	Pilihan kota dan judul wisata
Keluaran yang diharapkan	Wisata yang dicari dan daftar rekoemendasi
Hasil Pengujian	Berhasil



Gambar 5.2. Halaman Pencarian Cepat

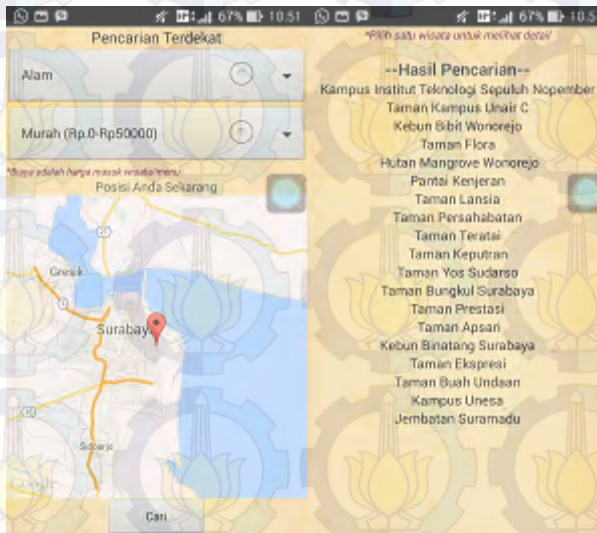
5.3.1.3. Pengujian Pencarian Wisata Terdekat

Skenario pengujian ini merupakan skenario pengujian untuk mengetahui fungsionalitas aplikasi pada fitur pencarian wisata terdekat dengan lokasi pengguna. Skenario pengujian dijelaskan pada Tabel 5.5. Tampilan pengujian diperlihatkan pada Gambar 5.3.

Tabel 5.5. Pengujian Melakukan Pencarian Wisata Terdekat

Nomor	PW03
Nama	Pencarian wisata terdekat
Use Case	F-0001-3
Tujuan	Daftar tempat wisata terdekat dengan lokasi pengguna aplikasi
Kondisi awal	Pilihan kategori dan biaya belum terpilih

	sedangkan koordinat lokasi pengguna sudah didapatkan
Skenario	<ol style="list-style-type: none"> 1. Pengguna memilih menu pencarian terdekat 2. Pengguna memilih kategori dan biaya masuk 3. Pengguna menekan tombol cari
Masukan	Pilihan kategori, pilihan biaya masuk dan koordinat lokasi pengguna
Keluaran yang diharapkan	Daftar tempat wisata terdekat dengan pengguna
Hasil Pengujian	Berhasil



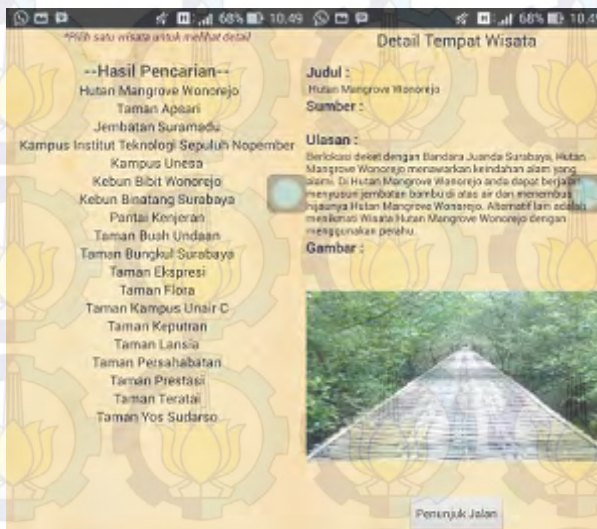
Gambar 5.3. Halaman Pencarian Wisata Terdekat

5.3.1.4. Pengujian Menampilkan Detail Wisata

Skenario pengujian ini merupakan skenario pengujian untuk mengetahui fungsionalitas aplikasi pada fitur melihat detail wisata. Skenario pengujian dijelaskan pada Tabel 5.6. Tampilan pengujian diperlihatkan pada Gambar 5.4.

Tabel 5.6. Pengujian Menampilkan Detail Wisata

Nomor	PW04
Nama	Menampilkan detail wisata
Use Case	F-0001-4
Tujuan	Menampilkan detail wisata
Kondisi awal	Pengguna sudah melakukan pencarian
Skenario	1. Pengguna memilih satu wisata untuk melihat detail wisata
Masukan	Judul wisata
Keluaran yang diharapkan	Aplikasi menampilkan detail wisata yang meliputi judul, ulasan, alamat sumber dan gambar.
Hasil Pengujian	Berhasil

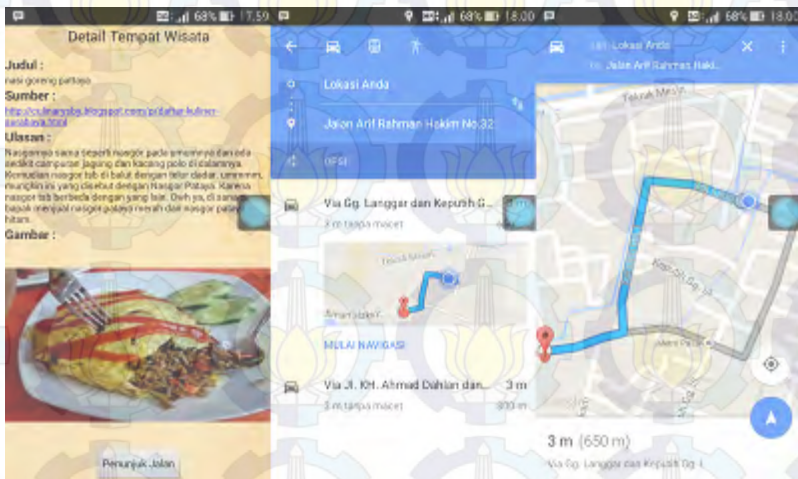
**Gambar 5.4. Halaman Detail Wisata**

5.3.1.5. Menampilkan Penunjuk Jalan

Skenario pengujian ini merupakan skenario pengujian untuk mengetahui fungsionalitas penunjuk jalan. Skenario pengujian dijelaskan pada Tabel 5.7. Tampilan pengujian diperlihatkan pada Gambar 5.5.

Tabel 5.7. Pengujian Menampilkan Penunjuk Jalan

Nomor	PW05
Nama	Penunjuk Jalan
Use Case	F-0002
Tujuan	Memberikan penunjuk jalan ke pengguna dari lokasi sekarang ini ke tempat wisata
Kondisi awal	Pengguna sudah melihat detail wisata
Skenario	1. Pengguna menekan tombol penunjuk jalan pada halaman detail wisata 2. Pengguna memilih jalan yang diinginkan
Masukan	Koordinat wisata yang terdapat pada detail wisata
Keluaran yang diharapkan	Aplikasi menampilkan penunjuk jalan dari lokasi pengguna ke lokasi wisata
Hasil Pengujian	Berhasil

**Gambar 5.5. Halaman antarmuka penunjuk jalan**

5.3.1. Pengujian Pengguna Admin

Pada pengujian pengguna aplikasi administrasi, diuji tentang kehandalan fungsi atau yang berkenaan dengan kebutuhan pengguna administrasi.

5.3.2.1. Melihat Wisata

Skenario pengujian ini merupakan skenario pengujian untuk mengetahui fungsionalitas melihat wisata. Skenario pengujian dijelaskan pada Tabel 5.8. Tampilan pengujian diperlihatkan pada Gambar 5.6.

Tabel 5.8. Pengujian Melihat Wisata Administrator

Nomor	PW06
Nama	Melihat wisata
Use Case	F-0006-1
Tujuan	Administrator dapat melihat wisata sesuai yang diinginkan
Kondisi awal	Pilihan provinsi, pilihan kota dan pilihan kategori belum dipilih
Skenario	1. Pengguna memilih menu pencarian 2. Pengguna memilih provinsi, kota dan kategori 3. Pengguna menekan tombol cari
Masukan	Pilihan provinsi, kota dan kategori
Keluaran yang diharapkan	Daftar wisata yang sesuai dengan pilihan administrator
Hasil Pengujian	Berhasil



Gambar 5.6. Halaman Melihat Wisata Administrator

5.3.2.2. Tambah Wisata Manual

Skenario pengujian ini merupakan skenario pengujian untuk mengetahui fungsionalitas menambah wisata manual. Skenario pengujian dijelaskan pada Tabel 5.9. Tampilan pengujian diperlihatkan pada Gambar 5.7 dan Gambar 5.8.

Tabel 5.9. Pengujian Tambah Manual

Nomor	PW07
Nama	Menambah wisata manual
Use Case	F-0006-2
Tujuan	Menambah data wisata secara manual
Kondisi awal	Semua masukkan belum terisi
Skenario	<ol style="list-style-type: none"> 1. Pengguna memilih menu tambah manual 2. Pengguna memilih dan memasukkan data wisata yang dibutuhkan pada halaman tambah wisata manual 3. Pengguna menekan tombol tambah
Masukan	Data wisata yang dibutuhkan
Keluaran yang diharapkan	Detail wisata yang telah ditambahkan
Hasil Pengujian	Berhasil

Tambah Wisata Manual
Kolom dengan * wajib diisi.

Provinsi	Jawa Timur
Kota	Surabaya
Kategori	Kuliner
Judul	Bebek Oven Mozzarella pak joss
Deskripsi	<p>Berawal dan gerobak yang keliling di sekitar perumahan menggunakan jingle. Menu best sellernya adalah bebek oven mozzarella. Bebek dipotong kecil-kecil, disajikan di hotplate, dioven dan disiram mozzarella di atasnya.</p>
Alamat Web Sumber	http://ceritaperut.com/user/BebekPakJoss Ukuran maksimal gambar 300px
Foto	Browse... No file selected. Harus bebek_pakjoss.png
Biaya	50000 <small>*Biaya masuk/marga berporal makanan</small>
Tempat	
Koordinat X	-7.289451
Koordinat Y	112.755617
<input type="button" value="Tambah"/> <input type="button" value="Batal"/>	

Gambar 5.7. Halaman Tambah Wisata Manual

Pencarian	Tambah Wisata	Pengaturan	Logout (sodik)
-----------	---------------	------------	----------------

[Home](#) » [Wisata](#) » bebek oven mozzarella pak joss

Detail Wisata bebek oven mozzarella pak joss

Idtempatwisata	459
Judul	bebek oven mozzarella pak joss
Link	http://ceritaperut.com/user/BebekPakJoss
Deskripsi	Berawal dan gerobak yang keliling di sekitar perumahan menggunakan jingle. Menu best sellernya adalah Bebek Oven Mozzarella. Bebek dipotong kecil-kecil, disajikan di hotplate, dioven dan disiram mozzarella di atasnya.
Foto	bebek_pakjoss.png
Biaya	Murah
Nominalbiaya	50000
Koordinatx	-7.2894510
Koordinaty	112.7556170

Gambar 5.8. Halaman Detail Wisata Tambah Manual

5.3.2.3. Tambah Wisata Dengan *Scraping Website* Lain

Skenario pengujian ini merupakan skenario pengujian untuk mengetahui fungsionalitas menambah wisata dengan *scraping website* lain. Skenario pengujian dijelaskan pada Tabel 5.10. Tampilan pengujian diperlihatkan pada Gambar 5.9 dan Gambar 5.10.

Tabel 5.10. Pengujian Menambah Wisata dengan Scraping

Nomor	PW08
Nama	Menambah wisata dengan scraping website lain
Use Case	F-0006-3
Tujuan	Menambahkan data wisata dengan scraping website lain
Kondisi awal	Semua masukkan belum terisi
Skenario	<ol style="list-style-type: none"> 1. Pengguna memilih menu tambah data website 2. Pengguna memilih dan memasukkan script wisata yang akan di scraping 3. Pengguna menekan tombol tambah
Masukan	Script website wisata yang dibutuhkan
Keluaran yang diharapkan	Detail wisata yang telah ditambahkan
Hasil Pengujian	Berhasil

Pencarian **Tambah Wisata** Pengaturan Logout (sedik)

Home » Tambah Web

Tambah Wisata Website

Kelom dengan * diisi secara manual.

Provinsi	Jawa Timur
Kota	Tulungagung
Kategori	Alam
Alamat Web Sumber	sya.com/tourism/tulungagung/ina/pathok_gebang.html
Judul	<h2> </h2>
Deskripsi	<p> </p>
Foto	http://www.eastjava.com/tourism/tulungagung/ina/
Biaya Masuk	0 *Diwajibkan masuk harga persori makanan
Tempat	:
Koordinat X	-8.302182
Koordinat Y	111.901330

Tambah Batal

Gambar 5.9. Halaman Tambah Wisata Dengan Scrapping Website Lain

Pencarian **Tambah Wisata** Pengaturan Logout (sedik)

Home » Wisata » pantai pathok gebang, tulungagung

Detail Wisata pantai pathok gebang, tulungagung

Idtempwisata	458
Judul	pantai pathok gebang, tulungagung
Link	http://www.eastjava.com/tourism/tulungagung/ina/pathok_gebang.html
Deskripsi	Pantai Pathok Gebang terletak di Desa Jenggulhango, Ken Tanggunggunung Tulungagung Jawa Timur. Pantai Pathok Gebang merupakan pantai yang masih alami, karena menjadi salah satu yang masih tersembunyi di pesisir Tulungagung. Tidak hanya akses jalan yang masih belum memadai, tapi juga karena lokasinya yang berada jauh dari pemukiman penduduk.
Foto	http://www.eastjava.com/tourism/tulungagung/images/pathok-gebang.jpg
Biaya	Murah
Nominalbiaya	0
Koordinatx	-8.3021820
Koordinaty	111.9013300

Kembali

Gambar 5.10. Halaman Detail Wisata Tambah Dengan Scrapping Website Lain

5.3.2.4. Mengelola Kota

Skenario pengujian ini merupakan skenario pengujian untuk mengetahui fungsionalitas mengelola kota. Skenario pengujian dijelaskan pada Tabel 5.11. Tampilan pengujian diperlihatkan pada Gambar 5.11.

Tabel 5.11. Pengujian Mengelola Kota

Nomor	PW10
Nama	Mengelola kota
Use Case	F0004
Tujuan	Mengelola kota
Kondisi awal	Menampilkan daftar kota
Skenario	1. Pengguna memilih menu pengaturan kota 2. Pengguna memilih provinsi yang akan diubah/hapus/memilih menu tambah kota
Masukan	Tidak ada
Keluaran yang diharapkan	Daftar wisata terbaru
Hasil Pengujian	Berhasil



Gambar 5.11. Halaman Mengelola kota

5.3.2.5. Mengelola Kategori

Skenario pengujian ini merupakan skenario pengujian untuk mengetahui fungsionalitas mengelola kategori. Skenario pengujian dijelaskan pada Tabel 5.12. Tampilan pengujian diperlihatkan pada Gambar 5.12.

Tabel 5.12. Pengujian Mengelola Kategori

Nomor	PW11
Nama	Mengelola kategori
Use Case	F0005
Tujuan	Mengelola kategori
Kondisi awal	Menampilkan daftar kategori
Skenario	<ol style="list-style-type: none"> 1. Pengguna memilih menu pengaturan kategori 2. Pengguna memilih kategori yang akan diubah/hapus/memilih menu tambah kategori
Masukan	Tidak ada
Keluaran yang diharapkan	Daftar kategori yang terbaru
Hasil Pengujian	Berhasil



Gambar 5.12. Halaman Mengelola Kategori

5.3.2.6. Mengelola Wisata

Skenario pengujian ini merupakan skenario pengujian untuk mengetahui fungsionalitas mengelola wisata. Skenario pengujian dijelaskan pada Tabel 5.13. Tampilan pengujian diperlihatkan pada Gambar 5.13.

Tabel 5.13. Pengujian Mengelola Wisata

Nomor	PW12
Nama	Mengelola wisata
Use Case	F-0006-4
Tujuan	Mengelola wisata
Kondisi awal	Menampilkan daftar wisata
Skenario	<ol style="list-style-type: none"> 1. Pengguna memilih menu pengaturan wisata 2. Pengguna memilih wisata yang akan dirubah/hapus/memilih menu tambah wisata
Masukan	Tidak ada
Keluaran yang diharapkan	Daftar wisata yang terbaru
Hasil Pengujian	Berhasil



Gambar 5.13. Halaman Mengelola Wisata

5.4. Pengujian Kegunaan

Selain melakukan pengujian fungsionalitas dengan metode *black box*, juga dilakukan pengujian kegunaan untuk menilai kegunaan perangkat lunak secara langsung kepada pengguna. Pengujian kegunaan dilakukan untuk mengetahui penilaian dan tanggapan dari pengguna terhadap sejumlah aspek dari perangkat lunak ini.

5.4.1. Kriteria Responden

Pengujian kegunaan dilakukan dengan melibatkan beberapa pengguna dari aplikasi ini. Para pengguna yang terlibat dalam pengujian ini selanjutnya disebut sebagai responden. Para responden diberikan keleluasaan untuk menjalankan aplikasi pada perangkat bergerak Android.

5.4.2. Lingkungan Pengujian Kegunaan

Masing-masing responden menjalankan perangkat lunak pada perangkat bergerak berbasis Android. Perangkat bergerak yang digunakan oleh responden terhubung dengan jaringan internet dengan versi minimum Android 4.2. (Jelly Bean) dan perangkat bergerak dengan layar minimum 4 inc.

5.4.3. Skenario Pengujian Kegunaan

Dalam melakukan pengujian kegunaan aplikasi, responden diminta untuk menjalankan aplikasi dan melakukan pencarian wisata berdasarkan daerah. Dalam memberikan penilaian terhadap aplikasi, responden mengisi formulir penilaian yang telah disediakan untuk pengujian ini. Formulir penilaian memiliki beberapa aspek penilaian secara kualitatif. Form survei Kuisisioner Pengujian Kegunaan terdapat pada lampiran A.

5.4.4. Daftar Responden

Terdapat 15 orang yang menjadi responden dalam pengujian kegunaan aplikasi. Daftar responden beserta keterangan perangkat bergerak yang digunakannya serta lokasi ketika menggunakan aplikasi terdapat pada Tabel 5.14.

Tabel 5.14. Daftar Responden Pengujian Kegunaan

No	Nama	Tipe Perangkat Bergerak	Versi Android	Lokasi
1.	Aichiril Hajri	Asus Zenfone 5	4.4.2 (KitKat)	Keputih, Surabaya
2.	Agus Triwibowo	Asus Zenfone 5	4.4.2 (KitKat)	Keputih, Surabaya
3.	Hizkia Adi Surya	Asus Zenfone 5	4.4 .2(KitKat)	Rungkut, Surabaya
4.	Zakaria A	Asus Zenfone 4	4.4.2 (KitKat)	Keputih, Surabaya
4.	Rizqi Ahmad Ramadhani	Samsung Galaxy Star	4.2 (Jelly Bean)	Keputih, Surabaya
5.	Mia Ambarwati	Samsung Galaxy Core 2	4.4.2 (KitKat)	Kediri
6.	Bramantyo	Lenovo A369	4.2.2(Jelly Bean)	Kampus ITS, Surabaya
7.	Bobby Kristya Nareswara	Lenovo A369	4.2.2 (Jelly Bean)	Kampus ITS, Surabaya
8.	Wildan Ibrahim Amin	Lenovo A369	4.2.2 (Jelly Bean)	Kampus ITS, Surabaya
9.	Adhi MS	Samsung Galaxy Fame	4.2(Jelly Bean)	Keputih, Surabaya
10.	Deni	Acer Z4	4.2 (Jelly Bean)	Keputih, Surabaya
11.	Muktar Redy S	Sony Xperia M2	4.4.2 (KitKat)	Keputih, Surabaya
12.	Arie Priyambada	Asus Zenfone 5	4.4.2 (KitKat)	Keputih, Surabaya
13.	Ayu Wandira	Lenovo A369	4.2.2 (Jelly Bean)	Landungsari, Malang
14.	Siti Munawaroh	Lenovo A369	4.2.2(Jelly Bean)	Rungkut, Surabaya

5.4.5. Hasil Pengujian Kegunaan

Pada penilaian pengujian kegunaan, terdapat empat pilihan nilai untuk setiap pertanyaan yaitu tidak baik, kurang baik, cukup baik, dan sangat baik. Untuk menghitung hasil pengujian, maka dilakukan pengubahan nilai tersebut menjadi nilai-nilai angka dan persentase seperti pada Tabel 5.15.

Tabel 5.15. Daftar Perubahan Penilaian Pada Pengujian Kegunaan

No	Nilai Pada Kuesioner	Nilai Angka	Persentase
1	Tidak Baik	1	25 %
2	Kurang Baik	2	50 %
3	Cukup Baik	3	75 %
4	Sangat Baik	4	100 %

Pengujian dibagi menjadi dua aspek yaitu antarmuka pengguna, dan kebutuhan fungsional aplikasi. Berikut ini adalah rekapitulasi penilaian dari hasil pengujian kegunaan.

5.4.5.1 Rekapitulasi Pengujian Aplikasi

Rekapitulasi pengujian meliputi hasil penilaian aspek antarmuka pengguna dan aspek kebutuhan fungsional.

Pengujian antarmuka menilai kemudahan penggunaan aplikasi dan pengujian kebutuhan fungsional menilai keberhasilan aplikasi dalam melakukan fungsi dan kegunaan bagi para penggunanya. Rekapitulasi penilaian pengujian aplikasi dapat dilihat pada Tabel 5.16

Tabel 5.16. Penilaian Kebutuhan Fungsional.

No.	Penilaian Antarmuka				Rata-rata
	Tidak	Kurang	Cukup	Sangat	
1	0	1	8	6	3,36
2	0	0	7	8	3,60
Nilai Akhir					3,48
No.	Penilaian Kebutuhan Fungsional				Rata-rata
	Tidak	Kurang	Cukup	Sangat	
1	0	0	8	7	3,40
2	0	0	9	6	3,40
3	0	0	5	10	3,60
4	0	0	9	6	3,40
5	0	0	7	8	3,53
Nilai Akhir					3,47

Berdasarkan Tabel 5.16 di atas, pengujian antarmuka aplikasi menunjukkan nilai rata-rata 3,48. Nilai tersebut menunjukkan bahwa aspek antarmuka sudah lebih dari cukup. Sedangkan pengujian kebutuhan fungsional menunjukkan nilai rata-rata 3,47. Nilai tersebut menunjukkan bahwa aspek fungsi dan aplikasi sudah lebih dari cukup.

5.4.5.2 Evaluasi Pengujian Fungsionalitas

Pengujian fungsionalitas yang telah dilakukan memberikan hasil yang sesuai dengan skenario yang telah direncanakan. Evaluasi pengujian untuk masing-masing fungsional aplikasi ini dijelaskan sebagai berikut.

1. Pengujian melakukan pencarian wisata sesuai dengan kriteria telah berjalan sesuai dengan harapan. Dibuktikan dengan pengujian kebutuhan fungsional pada no 1 dengan nilai rata-rata 3,40 yang menunjukkan lebih dari cukup.
2. Pengujian melakukan pencarian cepat dengan rekomendasi wisata yang hampir mirip telah berjalan sesuai dengan harapan. Dibuktikan dengan pengujian kebutuhan fungsional pada no 2 dengan nilai rata-rata 3,40 yang menunjukkan lebih dari cukup.

3. Pengujian melakukan pencarian wisata terdekat telah berjalan sesuai dengan harapan. Dibuktikan dengan pengujian kebutuhan fungsional pada no 3 dengan nilai rata-rata 3,60 yang menunjukkan lebih dari cukup.
4. Pengujian menampilkan detail wisata telah berjalan sesuai dengan harapan. Dibuktikan dengan pengujian kebutuhan fungsional pada no 4 dengan nilai rata-rata 3,40 yang menunjukkan lebih dari cukup.
5. Pengujian menampilkan penunjuk jalan dari posisi pengguna ke lokasi wisata telah berjalan sesuai dengan harapan. Dibuktikan dengan pengujian kebutuhan fungsional pada no 5 dengan nilai rata-rata 3,53 yang menunjukkan lebih dari cukup.

5.4.5.3 Evaluasi Pengujian Kegunaan

Untuk mengevaluasi pengujian kegunaan perangkat lunak, maka dilakukan rekapitulasi akhir. Rekapitulasi akhir menghasilkan nilai persentase terhadap aspek-aspek yang dinilai dalam pengujian kegunaan. Rekapitulasi akhir terdapat pada Tabel 5.17.

Tabel 5.17. Rekapitulasi Akhir Pengujian Kegunaan.

No.	Aspek Pengujian	Nilai Akhir	Persentase
1	Antarmuka Pengguna	3,48	87,00 %
2	Kebutuhan Fungsional	3,47	86,75 %

Dari Tabel 5.19 dapat dilihat bahwa aspek-aspek yang dinilai menghasilkan nilai-nilai persentase. Penilaian terhadap antarmuka menghasilkan persentase sebesar 87,00 %, penilaian terhadap kebutuhan fungsional menghasilkan persentase 86,75%, sehingga dapat disimpulkan bahwa dari segi kegunaan, aplikasi yang dibuat pada Tugas Akhir ini memberikan hasil yang lebih dari cukup.

5.4.6. Pengujian Perbandingan Kebutuhan Fungsional dengan Aplikasi WisataLokal dan Aplikasi Wisataku

Pada pengujian aplikasi WisataINA, aplikasi dibandingkan aplikasi lain. Perbandingannya adalah banyaknya kebutuhan fungsional yang dapat digunakan oleh pengguna dan kegunaan aplikasi bagi pengguna. Pengujian perbandingan dengan aplikasi WisataLokal dan aplikasi Wisataku. Skenario perbandingan kebutuhan fungsional dapat dilihat pada Tabel 5.18 dan hasil perbandingan dapat dilihat pada tabel 5.19.

Tabel 5.18. Skenario Perbandingan Pencarian

Nomor	A001
Nama	Pengujian perbandingan banyaknya kebutuhan fungsional dan kegunaan aplikasi bagi pengguna dengan aplikasi WisataLokal dan Wisataku
Usecase	F-0001-F0006
Kondisi awal	
Skenario	1. Pengguna mencoba semua fitur yang ada pada aplikasi WisataINA, Wisataku dan WisataLokal
Masukan	Opsional
Keluaran	Daftar wisata hasil uji coba
Hasil pengujian	Berhasil

Tabel 5.19. Hasil Perbandingan Aplikasi

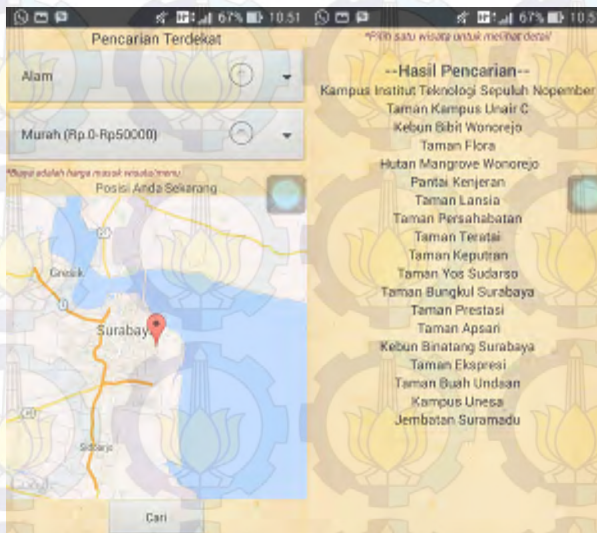
Fitur Pencarian	Wisata Daerah	Wisataku	WisataLokal
Pencarian cepat	Ditemukan	Ditemukan	Ditemukan
Rekomendasi Wisata	Ditemukan	—	—
Pencarian Sesuai Kriteria	Ditemukan	—	—
Pencarian Terdekat	Ditemukan	—	—
Detail Wisata	Ditemukan	Ditemukan	Ditemukan
Penunjuk Jalan	Ditemukan	Ditemukan	—

Bukti hasil perbandingan kebutuhan fungsional aplikasi dari Tabel 5.21 dapat dilihat pada gambar 5.14, 5.15, 5.16, 5.17, 5.18, 5.19, 5.20, 5.21, 5.22 dan 5.23

**Gambar 5.14. WisataINA Fitur Pencarian Sesuai Kriteria.**



Gambar 5.15. WisataINA Fitur Pencarian Cepat Dengan Rekomendasi



Gambar 5.16. WisataINA Fitur Pencarian Wisata Terdekat



Gambar 5.17. Wisataku Fitur Pencarian



Gambar 5.18. WisataLokal Fitur Pencarian



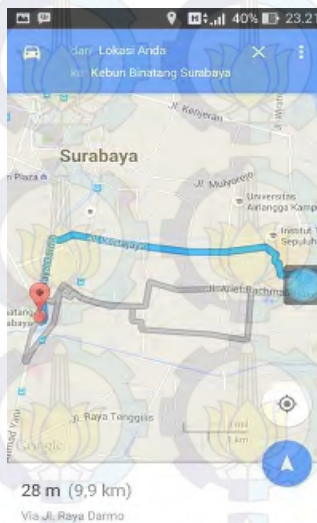
Gambar 5.19. WisataINA Detail Wisata



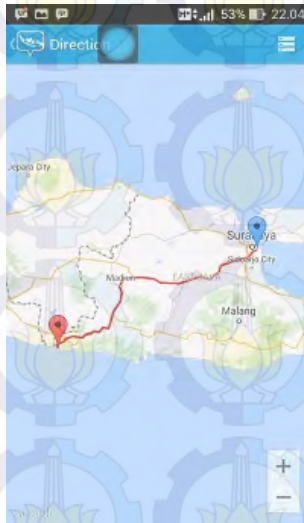
Gambar 5.20. Wisataku Detail Wisata



Gambar 5.21. WisataLokal Detail Wisata



Gambar 5.22. WisataINA Fitur Penunjuk Jalan



Gambar 5.23. Wisataku Fitur Penunjuk Jalan

BAB VI KESIMPULAN DAN SARAN

Pada bab ini diberikan kesimpulan yang diambil selama pengerjaan Tugas Akhir serta saran-saran tentang pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

6.1. Kesimpulan

Dari hasil selama proses perancangan, implementasi, serta pengujian aplikasi pencarian wisata daerah dapat diambil kesimpulan sebagai berikut:

1. Dengan menggunakan metode *scraping website*, data aplikasi WisataINA dapat diperoleh dari *website* wisata lain. Metode *scraping* akan mengekstrak isi data pada *website* sumber secara spesifik dan kemudian mengambil konten yang diperlukan untuk disimpan pada basis data. Dalam hal ini konten (data) yang diambil dari *website* sumber adalah judul, ulasan, gambar wisata. Implementasi metode *scraping* dilakukan pada *controller* pengaturan wisata di kelas tambah wisata *website*. Untuk mengakses *website* sumber dan sistem melakukan *scraping* dapat menggunakan peramban mozilla.
2. Dengan Metode *Analytic Hirachy Proses* (AHP), aplikasi WisataINA dapat memberikan rekomendasi wisata kepada pengguna ketika pengguna melakukan pencarian wisata. Perhitungan metode AHP menggunakan dua aspek yaitu aspek kategori wisata dan aspek biaya masuk wisata. Dari perhitungan didapatkan nilai bobot kategori sebesar 75% dan bobot biaya masuk sebesar 25% dengan nilai CR –tak terhingga, maka konsisten. Sehingga rekomendasi wisata lebih mementingkan kategori wisata dari pada biaya masuk wisata. Implementasi metode AHP dalam bentuk *query* yang terletak pada *web service*. *Web service* adalah penghubung antar aplikasi perangkat bergerak dengan basis data sebagai alat pertukaran data.

3. Dengan menggunakan Rumus Haversine, aplikasi WisataINA dapat memberikan rekomendasi wisata terdekat dengan lokasi pengguna sekarang ini. Perhitungan Rumus Haversine dengan menggunakan dua titik koordinat untuk membuat sebuah lingkaran pada permukaan bumi. Koordinat pengguna merupakan pusat lingkaran dan koordinat kedua diambil secara acak sesuai dengan radius yang ditentukan oleh pembuat aplikasi. Jarak titik pusat dan titik kedua adalah jari-jari lingkaran. Rekomendasi wisata adalah wisata yang letak koordinatnya dalam lingkaran yang terbuat dari Rumus Haversine. Implementasi Rumus Haversine dalam bentuk *query* yang terletak pada *web service*.
4. Dengan memanfaatkan aplikasi Google Maps, aplikasi wisata WisataINA dapat membantu pengguna menunjukkan jalan dari posisi pengguna sekarang ini ke lokasi wisata yang dituju dan sekaligus dapat digunakan sebagai *navigasi* dalam proses perjalanan menuju lokasi wisata.

6.2. Saran

Berikut saran-saran untuk pengembangan dan perbaikan sistem di masa yang akan datang berdasarkan hasil analisa perancangan dan implementasi. Diantaranya adalah sebagai berikut:

1. Perlu adanya kebutuhan fungsional pencarian kompleks yang pengguna dapat memilih beberapa daerah, beberapa kategori dan beberapa range biaya sehingga pengguna dapat lebih leluasa melakukan pencarian wisata.
2. Pada perkembangan selanjutnya, aplikasi dapat di tambahkan fitur pencarian hotel. Tujuannya agar pengguna selain mencari wisata dapat juga mendapatkan penginapan di lokasi wisata.
3. Apabila fitur pencarian hotel sudah ada, dapat ditambahkan fitur penanganan transaksi hotel, sehingga pengguna akan sangat dimudahkan dalam melakukan perjalanan wisata.

DAFTAR PUSTAKA

- [1] Ika, “Pengapalan Smartphone Android Tembus 1 Miliar Unit,” Telecommunication Trendsetter, 1 January 2015. [Online]. Available: <http://telsetnews.com/95395/pengapalan-smartphone-android-tembus-1-miliar-unit/>. [Diakses 11 Juny 2015].
- [2] Kementerian Pariwisata Republik Indonesia, “Jawa Timur,” Kementerian Pariwisata Republik Indonesia, 2013. [Online]. Available: <http://www.indonesia.travel/id/discover-indonesia/region-detail/34/jawa-timur>. [Diakses 11 Juny 2015].
- [3] IndonesiaWebpromotion, “eastjava.com,” IndonesiaWebpromotion.com, [Online]. Available: <http://www.eastjava.com/>. [Diakses 20 Oktober 2014].
- [4] json.org, “Introducing JSON,” json.org, [Online]. Available: <http://www.json.org/>. [Diakses 15 Februari 2013].
- [5] D. Arganto, “Aplikasi Mobile Rekomendasi Peencarian Obat dan Apotek memanfaatkan Google Maps,” Institut Teknologi Sepuluh Nopember, Surabaya, 2014.
- [6] V. Pradita, “Keunggulan dan Kelemahan MySQL,” 2011. [Online]. Available: <http://worldofmilanisti.blogspot.com/2012/10/jenis-jenis-protocol-dan-fungsinya-pada.html>. [Diakses 11 Juny 2015].
- [7] J. Ward, “Working With The Scraped Data [Part 2],” Jacob Ward, 21 January 2013. [Online]. Available: <http://www.jacobward.co.uk/working-with-the-scraped-data-part-2/>. [Diakses 20 Oktober 2014].
- [8] E. Darmanto, N. Latifah dan N. Susanti, “PENERAPAN METODE AHP (ANALYTHIC HIERARCHY PROCESS) UNTUK MENENTUKAN KUALITAS GULA TUMBU,” *Jurnal SIMETRIS ISSN: 2252-4983*, vol. V, no. 1, pp. 75-82,

2014.

- [9] Aingindra, “Android Adalah – Pengertian Android – Sistem Operasi,” 23 Desember 2013. [Online]. Available: <http://www.aingindra.com/android-adalah-pengertian-android-sistem-operasi.html>. [Diakses 1 July 2015].
- [10] F. R. Sari dan D. I. Sensuse, “PENERAPAN METODE ANALYTIC HIERARCHY PROCESS DALAM SISTEM PENUNJANG KEPUTUSAN UNTUK PEMILIHAN ASURANSI,” *Jurnal Sistem Informasi MTI-UI*, ISBN 1412-8896, vol. IV, no. 2, pp. 100-109, 2012.
- [11] B. A. Minartiningtyas, “Prinsip Dasar Analytic Hierarchy Process (AHP),” 1 June 2013. [Online]. Available: <http://informatika.web.id/prinsip-dasar-analytic-hierarchy-process-ahp.htm>. [Diakses 2 Maret 2015].
- [12] D. N. Kirom, Y. Bilfaqih dan R. Effendie, “Sistem Informasi Manajemen Beasiswa ITS Berbasis Sistem Pendukung Keputusan Menggunakan Analytical Hierarchy Process,” *JURNAL TEKNIK POMITS*, vol. 1, no. 1, pp. 1-6, 2012.

LAMPIRAN A – FORMULIR PENILAIAN PENGUJIAN KEGUNAAN

Kuesioner Pengujian Kegunaan Aplikasi Perangkat Bergerak Pencarian Wisata Berdasarkan Daerah Berbasis Android

Data Responden

Tanggal :
Nama :
Profesi :

Lingkungan Pengujian

Tipe Ponsel :
Versi Android :
Lokasi Pengguna :

A. Antarmuka Pengguna

1. Apakah aplikasi ini memiliki antarmuka pengguna yang menarik?
a. Tidak b. Kurang c. Cukup d. Sangat
2. Apakah antarmuka aplikasi ini mudah dipahami?
b. Tidak b. Kurang c. Cukup d. Sangat

B. Kebutuhan Fungsional Aplikasi

1. Apakah aplikasi ini sudah dapat melakukan pencarian wisata berdasarkan kriteria dengan baik?
a. Tidak b. Kurang c. Cukup d. Sangat
2. Apakah aplikasi ini sudah dapat melakukan pencarian wisata dengan memasukan judul wisata dari pengguna dan memberikan rekomendasi wisata dengan baik?
a. Tidak b. Kurang c. Cukup d. Sangat

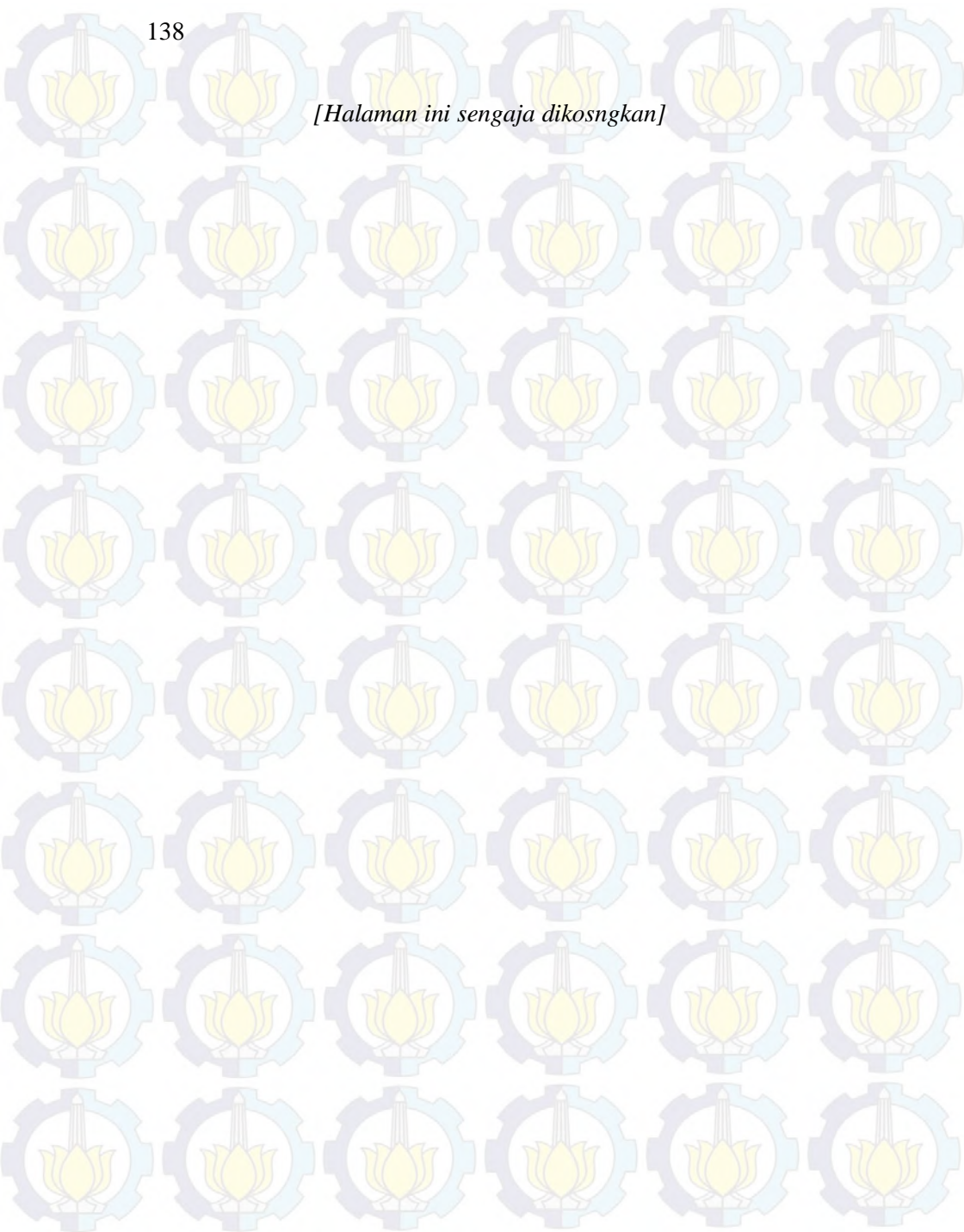
3. Apakah aplikasi ini sudah dapat melakukan pencarian wisata terdekat dengan lokasi pengguna dengan baik?
a. Tidak b. Kurang c. Cukup d. Sangat
4. Apakah aplikasi ini sudah dapat menampilkan informasi detail wisata dengan baik?
a. Tidak b. Kurang c. Cukup d. Sangat
5. Apakah aplikasi ini sudah dapat menampilkan informasi penunjuk jalan dari posisi pengguna sekarang ini ke lokasi tempat wisata dengan baik?
a. Tidak b. Kurang c. Cukup d. Sangat

LAMPIRAN B – IMPLEMENTASI SURVEI KRITERIA BIAYA

Tabel 5.23 Hasil Survei Kriteria Biaya				Tabel 5.24 Identitas Responden Survei Biaya			
Timestamp	Kriteria MURAH	Kriteria SEDANG	MAHAL	Nama Responden	No Hp Responden		
3/26/2015 17:28:23	Rp. 0 - Rp. 50000	Rp. 0 - Rp. 50000	Rp. 50000 - Rp. 150000	Ramadhan Wijayanto	+628983838099		
3/26/2015 17:31:28	Rp. 0 - Rp. 50000	Rp. 50001 - Rp. 150000	Lebih dari Rp. 150000	Aldo	-		
3/26/2015 19:00:16	20000	Rp. 0 - Rp. 50000	Lebih dari Rp. 150000	Tommy Anandra	85730157000		
3/26/2015 19:07:18	Rp. 0 - Rp. 50000	Rp. 50001 - Rp. 150000	Lebih dari Rp. 150000	Arie	82226242624		
3/26/2015 19:27:04	20000	Rp. 0 - Rp. 50000	Rp. 50000 - Rp. 150000	Aida	085948xxxx		
3/26/2015 19:27:48	Rp. 0 - Rp. 50000	Rp. 0 - Rp. 50000	Rp. 50000 - Rp. 150000	Arif Fauzi Aulia U R	85642221841		
3/26/2015 20:48:30	Maksimal 20000	10000	Rp. 50000 - Rp. 150000	Arif	85733638386		
3/27/2015 1:23:27	Rp. 0 - Rp. 50000	Rp. 50001 - Rp. 150000	Lebih dari Rp. 150000	erik wir			
3/27/2015 1:34:20	Rp. 50000 - Rp. 150000		Lebih dari Rp. 150000	DEVIRA PUSPA A	82214931214		
3/27/2015 6:56:10	Rp. 0 - Rp. 50000	Rp. 50001 - Rp. 150000	Lebih dari Rp. 150000	wilda	85646462770		
3/27/2015 9:40:01	Rp. 0 - Rp. 50000	Rp. 50001 - Rp. 150000	Rp. 50000 - Rp. 150000	ainasdi	81990924485		
3/27/2015 12:25:54	Rp. 0 - Rp. 50000	Rp. 50001 - Rp. 150000	Lebih dari Rp. 150000	Muh Fadjar R			
3/27/2015 15:42:11	Rp. 0 - Rp. 50000	Rp. 50001 - Rp. 150000	Lebih dari Rp. 150000	Raditya	85236557683		
3/28/2015 6:00:30	Rp. 0 - Rp. 50000	Rp. 50001 - Rp. 150000	Lebih dari Rp. 150000	nana	85730063595		
3/28/2015 12:13:14	Rp. 0 - Rp. 50000	Rp. 50001 - Rp. 150000	Lebih dari Rp. 150000	ayu	87859309075		
3/28/2015 14:11:04	Rp. 0 - Rp. 50000	Rp. 50001 - Rp. 150000	Lebih dari Rp. 150000	rizal			
4/8/2015 10:14:10	Rp. 0 - Rp. 50000	Rp. 50001 - Rp. 150000	Lebih dari Rp. 150000	Defria	85735442882		
4/11/2015 14:15:17	Rp. 0 - Rp. 50000	Rp. 50001 - Rp. 150000	Lebih dari Rp. 150000	syuaah	89684838027		
				Mahal	Jumlah		
Jumlah Rp. 0 - Rp. 50000				Jumlah Lebih dari Rp. 150000	#REF!		
Nilai yang lain				Nilai yang lain	10		
Total				Total	47	47	
Skala Kategori Biaya							
Murah							
Sedang							
Mahal							

Gambar 6.1. Implementasi Survei Kriteria Biaya

[Halaman ini sengaja dikosngkan]



LAMPIRAN C – IMPLEMENTASI SURVEI PRIORITAS KATEGORI/BIAYA MASUK

Timestamp	Wisata/biaya masuk	Jumlah	Nama Responden	Email Reponden
3/31/2015 12:55:41	Wisata	1	Adithya Agung Winoto	aditwinoto@gmail.com
3/31/2015 12:58:58	Biaya	0	ahmad tohir wijaya	ahmadtohir71@gmail.com
3/31/2015 12:58:28	Wisata	1	Mochamad Yasin	yasin_dobhe@yahoo.co.id
3/31/2015 13:00:14	Biaya	0	Muhamad Amiudin R	muara.mail@gmail.com
3/31/2015 13:03:20	Wisata	1	Nafisah	nafisahsugeng@gmail.com
3/31/2015 13:05:01	Wisata	1	Linda Nur Faizah	ljjahjouuw@yahoo.com
3/31/2015 13:05:44	Biaya	0	Bambang dwi	bams.dante@icloud.com
3/31/2015 18:42:32	Wisata	1	arie	arie.priyambadha@gmail.com
3/31/2015 22:52:20	Wisata	1	Eriska	d.eriska73@gmail.com
4/4/2015 17:14:01	Wisata	1	halimatus sakdiyah	diahlima16@gmail.com
4/4/2015 18:34:23	Wisata	1	adriyanra	yanra_adri@gmail.com
4/8/2015 10:12:41	Biaya	0	Detria	deiska17@gmail.com
4/8/2015 11:08:58	Wisata	1	Antonio Cahyadi L	liem.antonio@gmail.com
4/8/2015 12:19:05	Biaya	0	devia	devia_wandasan@yahoo.com
4/8/2015 12:59:28	Wisata	1	Ayu	wandiramanis@yahoo.co.id
4/8/2015 13:51:44	Wisata	1	Danis	Yordanis10.its@gmail.com
4/9/2015 10:33:56	Biaya	0		
4/9/2015 15:22:29	Biaya	0	Nurul Qamar	nurulqamar.aceh@gmail.com
4/11/2015 14:17:28	Wisata	1	syuaibah	
4/11/2015 14:24:03	Wisata	1	M Rizal Syah Karim	syah_karim@hotmail.com
4/11/2015 14:45:42	Wisata	1	Nur Arifin	nurarifin2010@gmail.com
4/11/2015 14:46:46	Wisata	1		
4/11/2015 14:50:08	Biaya	0	Niken Ayuning Tias	enchuzelek@yahoo.com
4/11/2015 15:01:53	Wisata	1	syahrul	syahrulmnf171@gmail.com
4/11/2015 15:11:23	Tempat Wisata	1	Lisa	Lisaranmiami@gmail.com
4/11/2015 15:17:12	Biaya	0	Iqhma Tofanie	fani_reyz@yahoo.com
4/11/2015 15:18:05	Wisata	1	Uyya	uyuyuyyya@gmail.com
4/11/2015 15:25:29	Wisata	1	Rizqi Ilman	rizkiilman58@yahoo.com
4/11/2015 15:42:05	Wisata	1	ulya	ulya.harier@gmail.com
4/11/2015 15:58:19	Wisata	1	Rosna Malika	rosna23@gmail.com
4/11/2015 16:16:28	Biaya	0	Reza Tianto	rzatianto@gmail.com
	Tempat Wisata	25		
	Biaya yang dikeluarkan	11		
	Jumlah	36		

Gambar 6.2. Survei Prioritas Kategori/Biaya Masuk

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Penulis, Ahmad Sodik, lahir di Kota Tulungagung pada tanggal 20 Oktober 1990, merupakan anak ke-9 dari sembilan bersaudara dan dibesarkan di Kabupaten Tulungagung, Jawa Timur. Penulis telah menempuh pendidikan di SDN Gendingan II, Kedungwaru, Tulungagung (1997-2003), SMPN 2 Kedungwaru, Tulungagung (2003-2006), MAN 1 Tulungagung (2006-2010), pada waktu menempuh pendidikan MAN, penulis pernah mengalami kecelakaan yang diharuskan berhenti belajar selama 1 tahun. Pada tahun 2010, penulis memulai pendidikan S1 jurusan Teknik Informatika Fakultas Teknologi Informasi di Institut Teknologi Sepuluh Nopember Surabaya, Jawa Timur. Selama masa perkuliahan di Teknik Informatika ITS, penulis aktif dalam kegiatan keorganisasian, diantaranya Himpunan Mahasiswa Teknik Computer-Informatika (HMTIC), Jama'ah Masjid Manarul Ilmi (JMMI) dan kepanitiaan acara Schematics periode 2012. Dalam menyelesaikan pendidikan sarjana, penulis mengambil bidang minat Rekayasa Perangkat Lunak (RPL) dan karena perubahan penjurusan, penulis berubah bidang minat ke Manajemen Informasi (MI). Penulis dapat dihubungi melalui email di sodik10its@gmail.com.